WebSphere® software

IBM

WebSphere Portal for z/OS Version 6

Creating a WebSphere Portal environment on z/OS

Implementing high availability and security

Configuring and using Web Content Management

Alex Louwe Kooijmans
Egide van Aerschot
Nigel J Davies
Doris Fiorentino
John T Gates
Nicole Hargrove
Paul Houde
Foulques de Valence

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Portal for z/OS Version 6

October 2007

SG24-7459-00

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

First Edition (October 2007)

This edition applies to IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 and WebSphere Application Server for z/OS Version 6.02.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- Redbooks (logo) ®
- iSeries®
- z/OS®
- zSeries®
- AIX®
- Cloudscape™
- CICS®
- Domino®
- DB2 Universal Database™
- DB2®
- GDPS®
- HiperSockets™
- Info Plus™
- IBM®
- IMS™
- Lotus Notes®
- Lotus®
- MVS™
- Notes®
- Parallel Sysplex®
- Rational®
- Redbooks®
- RACF®
- Sametime®
- SmartSuite®
- System z™
- Tivoli®
- WebSphere®
- Workplace™
- Workplace Forms™
- Workplace Web Content Management™

The following terms are trademarks of other companies:

Java, Java DataBase Connectivity, JumpStart, JDBC, JSP, JVM, J2EE, J2SE, Sun Java, Ultra, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Internet Explorer, Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

A portal is one of the most important components in a Service Oriented Architecture (SOA). IBM® WebSphere® Portal is available for a variety of platforms, including z/OS® and z/Linux®. In this book we discuss IBM WebSphere Portal Enable for z/OS Version 6.0.0.1, which combines the rich functionality brought by IBM WebSphere Portal for Multiplatforms with the Qualities of Service provided by the z/OS platform to provide a robust computing platform. The information in this book is based on experiences gained during an ITSO Proof of Concept in which we installed and configured WebSphere Portal on z/OS in a highly available and secure environment. It is targeted to those who need to implement Portal in System z environments.

The team that wrote this IBM Redbooks publication

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

Alex Louwe Kooijmans is a Project Leader with the International Technical Support Organization (ITSO) in Poughkeepsie, NY. He specializes in WebSphere, Java™ and SOA on System z™ with a focus on integration, security, high availability, and application development. Previously Alex worked as a Client IT Architect in the Financial Services sector with IBM in The Netherlands, advising financial services companies on IT issues such as software and hardware strategy and on demand. He has also worked at the Technical Marketing Competence Center for zSeries® and Linux in Boeblingen, Germany, providing support to customers implementing Java and WebSphere on zSeries. From 1997 to 2002, Alex completed a previous assignment with the ITSO, managing various IBM Redbooks® projects and delivering workshops around the world.

Egide van Aerschot holds an engineering degree in Electricity and Nuclear Physics from the University of Leuven, Belgium. He joined IBM in 1967 and was responsible for many computer installations related to Tele-Processing and Database Management in Belgium. In 1997 he moved from IBM Belgium to IBM France, where he works as an architect and consultant at the IBM Program Support Center in Montpellier. Since 1997, he has specialized in Java and WebSphere applications mainly on z/OS systems, and participated in many projects related to the Internet. Egide is co-owner of the patent “Methods, systems, program product for transferring program code between computer processes”.

Nigel J Davies is an IT Architect working in Sydney, Australia for IBM Global Business Services. He earned his Masters degree in Chemistry at Oxford University in England. He has more than 25 years of experience in the IT industry, and has previously worked as a programmer, designer, business analyst, DBA, and project manager. Nigel is currently certified in DB2® Universal Database™ for z/OS and IBM Data Integration. He has published articles about DB2 performance for the UK trade journal Info Plus™ and co-authored the first IBM Redbook publication about DB2 Information Integrator in 2003. He has worked for IBM Australia for 11 years and with WebSphere products for the last three years.

Doris Fiorentino is an Advisory Software Engineer working in the USA. She holds a BA in Mathematics from Seton Hill University, and is working on earning an MBA from DeSales University. She has worked for IBM for more than 10 years, and focused on Domino® for the Lotus® brand of Software Group. Currently she is the team lead for the Domino on zSeries Level 2 support team, as well as for the Portal on zOS Level 2 support team. She is an IBM Certified System Administrator for Lotus Notes® and Domino 6/6.5 and 7. Doris has written several white papers on Domino for zOS topics, and helped to write several Buying and Selling Guides for Domino on zSeries and other platforms.

John T Gates is a Senior System z IT Architect who has worked at IBM for 25 years. He has been involved in all phases of Product Design, Development, Test, Service, and Services. His areas of expertise include z/OS, TCP/IP and SNA, and Web-based products (WebSphere and Portal.) John currently leads a team of software professionals who work with customers to exploit System z technologies in their enterprises. He holds a BS in Computer Information Systems and is a frequent speaker at customer and industry events.

Nicole Hargrove is an accredited Senior I/T Specialist on the Advanced Technical Support team supporting the Americas from the Washington System Center in Gaithersburg, Maryland. She has 12 years of experience in infrastructure and application design and deployment, and has worked at IBM for 10 years. Her areas of expertise include WebSphere Portal and Application Server Families on distributed as well as mainframe platforms and Rational® Application Developer. She is IBM Certified as a Systems Expert in WebSphere Application Server, IBM Certified as a Systems Administrator in WebSphere Portal Server, as well as IBM Certified as an Associate Developer in WebSphere Studio Application Developer. She holds a BS with a double major in Accounting and Information Systems from Old Dominion University, and an MS in Advanced Technology and eCommerce from Johns Hopkins University. She co-authored the IBM Redbooks publication *WebSphere Version 5 Application Development Handbook*.

Paul Houde is a Systems Management Integrator working in Southbury, CT, USA. He holds a double major from Fairfield University in Information Systems and Marketing. In 2002, he earned an MBA from Quinnipiac University. He started his IBM career working for Global Services on a team that supported DB2 on the z/OS platform. In his current assignment, Paul focuses on the installation, configuration, and maintenance of several products under the WebSphere brand including WebSphere Application Server, WebSphere Portal, and WebSphere Commerce running on most operating systems (z/OS, AIX®, Windows® and Linux).

Foulques de Valence is a System z Security IT Architect with IBM STG and is currently based in Poughkeepsie NY, USA, where he works for the Lab Services team. Previously, he was a Web infrastructure IT Architect in France focusing on SOA and z/OS. Foulques taught end-to-end security solutions for WebSphere on z/OS worldwide. He is a co-author of several IBM Redbooks publications dealing with security and WebSphere Application Server for z/OS. He received a Masters in Computer Science and Engineering from Ensimag in France. He furthered his education at the State University of New York at Buffalo and at Stanford University, CA, USA.

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

David Bennin, Richard Conway, Al Schwab
International Technical Support Organization, Poughkeepsie Center

Dr. Irene Stahl
IBM Development Lab for WebSphere Portal, Boeblingen

Greg Gasper, Paul Kelsey, Terry Swope
IBM Development Lab for WebSphere Portal, Raleigh

Stanley Tzueing Shieh
IBM Lotus Worldwide Technical Sales

Joe Breal, Theresa Smit
IBM Lotus Technical Sales Team for Web Content Management

Kevin Senior
IBM SWG Lab Services Europe, WebSphere/CICS® on z/OS

Tom Hackett
IBM STG Technical Sales Support, Service Oriented Architecture

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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 IBM Redbooks to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

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

  `ibm.com/redbooks`

- Send your comments in an e-mail to:

  `redbooks@us.ibm.com`

- Mail your comments to:

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

1

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1

In this chapter we introduce the IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 product. We start by listing the features and functions that are included with the product, as well as optional components that may be configured with Portal.

We continue with a discussion of the prerequisite and corequisite products, along with their required service levels. We discuss how Portal fits into a SOA architecture, and why Portal and SOA are a good fit for z/OS and System z. We present common Portal architectures and implementations, and highlight those which we use for further discussion later in this book.

We conclude with a discussion of why a Portal deployment on z/OS makes sense, and what the major advantages are of creating a portal infrastructure based on System z and z/OS.

1.1 WebSphere Portal V6.0

In this section we explain the key features, functions, and benefits afforded by the WebSphere Portal product. We start with a review of the key enhancements introduced by the base WebSphere Portal V6.0 product. We then explain the additional features added in the IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 product.

> **Note:** The focus of this part of the introductory chapter is a description of IBM WebSphere Portal Enable for z/OS Version 6.0.0.1. Any discussion of the Portal for z/OS product would not be complete, however, without first discussing the features and functions in the base IBM WebSphere Portal V6 product.
>
> Accordingly, this section of the book borrows extensively from an article entitled "What's new in WebSphere Portal Version 6?", which can be found at:
>
> `http://www-128.ibm.com/developerworks/WebSphere/libary/techarticles/0607_hepper/0607_hepper.html`
>
> We thank the authors of this paper:
>
> - Stefan Hepper (sthepper@de.ibm.com), WebSphere Portal Programming Model Architect, IBM
> - Stefan Liesche (liesche@de.ibm.com), Senior Technical Staff Member, WebSphere Portal and Workplace™ Foundation Lead Architect, IBM
> - Gregory Melahn (melahn@us.ibm.com), Senior Technical Staff Member, WebSphere Portal and Workplace Content Architect, IBM
> - Dr. Thomas Stober (tstober@de.ibm.com), Software Architect, WebSphere Portal Development, IBM

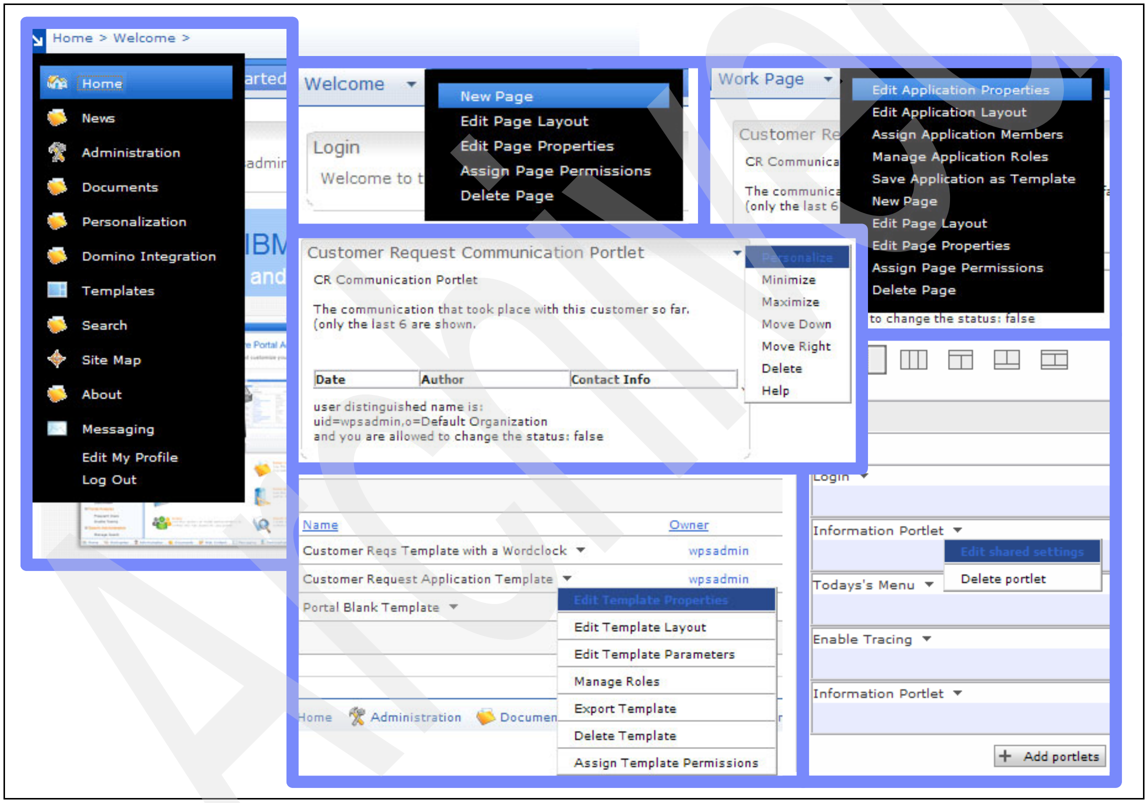
1.2 WebSphere Portal V6.0 key enhancements

WebSphere Portal V6 is a key part of the IBM *Service Oriented Architecture* (SOA) strategy. This version advances the state of the art in several key areas. It improves efficiency and productivity through a vastly improved user experience with additional personalization capabilities. Application and content creation is restructured and improved through the concept of composite applications.

There are new options for high availability and continuous operations, including support for multiple security realms and an enhanced database configuration. Portal V6 continues the commitment to open standards by further enriching support for key initiatives such as JSRs 168, 170, and WSRP. Portlet creation is enhanced with the use of the IBM Portlet Factory.

1.2.1 User interface enhancements

WebSphere Portal V6 has a significantly reworked user interface. The “look and feel” now has a modern, state-of-the-art design. As shown in, Figure 1-1it includes a drop-down menu, providing access to the main functional areas of the portal such as administration, content management, templating, and so on. Menu choices are offered within context making the new portal experience very intuitive.

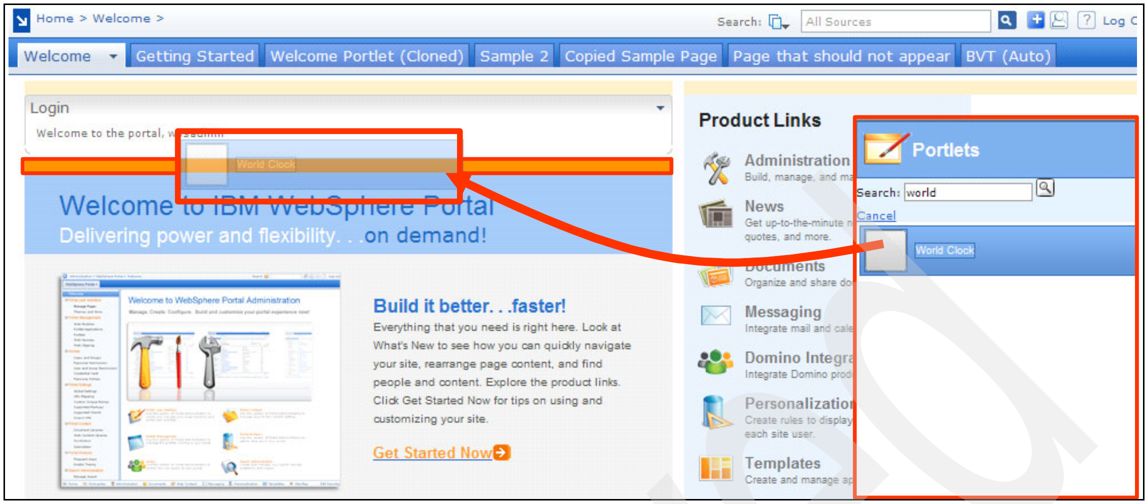


The communication that took place with this customer so far.
(only the last 6 are shown.

user distinguished name is:
uid=wpsadmin,o=Default Organization
and you are allowed to change the status: false

Figure 1-1 Main portal drop-down menu and contextual menus

The new Portlet Pallete, shown in Figure 1-2 on page 4, makes the creation and design of portal pages simple. The palette offers a list of components from which you can drag and drop directly onto your page. Using the palette simplifies the task of placing portlets on a page and makes the design of pages much easier.



Welcome to the portal, wpsadmin

Everything that you need is right here. Look at What's New to see how you can quickly navigate your site, rearrange page content, and find people and content. Explore the product links. Click Get Started Now for tips on using and customizing your site.

Figure 1-2 Drag and drop: Adding portlets to a page using the Portlet Palette

1.2.2 Composite applications and templates

Composing applications out of components is much more efficient than creating them from scratch. Placing complex business logic within portlets, however, can be difficult. Consider the steps necessary to build an application from a set of portlets:

1. Deploy the individual component parts.
2. Arrange the parts on the system.
3. Define the interaction models and access controls according to the business requirements specified.

Unfortunately, in order to complete these steps with any degree of success, you need to have multiple interactions with various disciplines. But now there is a better way to accomplish this task.

WebSphere Portal V6 introduces the concept of *composite applications*. Business personnel and application designers can easily assemble complex composite applications using components such as portlets, processes, and other artifacts.

Composite applications are a key tool in quickly realizing business value in an SOA. WebSphere Portal encourages those persons with unique skills and abilities to define, create, and manage their own composite applications within an environment that is comfortable for them to use.

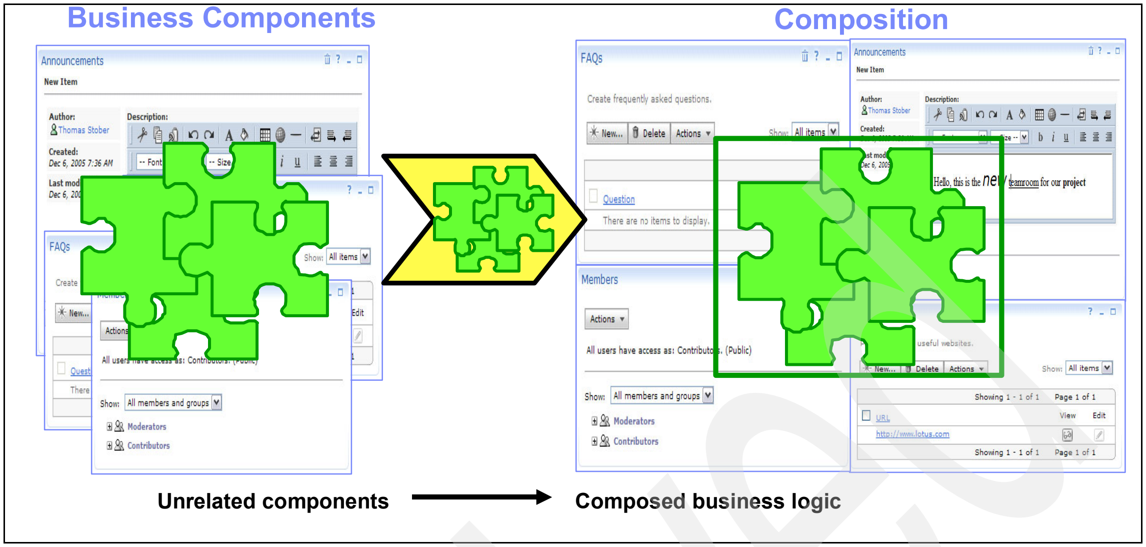


Figure 1-3 - Composing business logic from components

1.2.3 Personalization improvements

WebSphere Portal lets you *personalize* content for different groups of users. You can easily enable administration based on attributes. In previous versions, Portal lets you customize the content that a user saw, based on the user's role. In WebSphere Portal Version 6, you can define rules to modify the content a user sees based on the current request and a set of rules that apply to this request.

With WebSphere Portal 6, you can define and create groups of rules called *policies*. Policies can be used along with other personalization features and templates to enforce unique behaviors that affect a user's portal experience. For example, you could choose to enforce a different mail size quota for different classes of logged-on users.

1.2.4 Programming model changes

The WebSphere Portal V6 programming model is based on the JSR 168 Portlet specification. WebSphere Portal Version 6 adds substantial features beyond the capabilities provided for in the base JSR 168 specification.

Your ability to move portlets from one runtime to another depends largely upon how much of the programming model you choose to exploit in your application.

Improvements in WebSphere Portal version V6 include:

- Composite applications

  Create business applications and templates as previously described.

- Themes and skins extension points

  These are used to customize the default themes and skins.

- Drag and Drop

  Define your own drag and drop points within your application templates, portlets, themes, and skins.

- Policies

  Create policies using templates, rules, and personalization capabilities previously described.

- Support for Edit Default mode

  Distinguish between default settings used by a shared portlet among different classes of users.

- Advanced portlet URL generation

  Create URLs that point to other pages and portlets using the new URL generation SPIs.

- Additional portlet model and state SPIs

  Create aggregated meta data.

- Search

  Use a common search API for the various search engines available from IBM.

- Workflow

  Create your own objects for processing by the Workflow Builder tool.

1.2.5 Content and Web Content Management

IBM Workplace Web Content Management™ is included and licensed for use with WebSphere Portal.

The Web Content Management function has been completely rebased on the common *Java Content Repository* (JCR) that is included with WebSphere Portal Version 6. This repository, based on the JSR 170 open standard, is the same repository used for Portal documents and personalization rules. It enables the Web Content Management component to provide greater levels of scalability and performance than was possible in previous versions of WebSphere Portal.

Management of Web content is now easier because nodes can share the same repository allowing full clustering in both the authoring and production environments. Logging and caching for Web Content Management is now shared in a common implementation with WebSphere Portal.

Content authors and end users can now more easily find content through enhanced integration with Portal search capabilities. Portal search collections may now include content from the Web content repository.

The WebSphere Portal Document Manager now lets you access documents directly from your desktop Windows environment or from Windows Office products. Document libraries appear as nodes in the Network Places view of the Windows Explorer, giving easy access to common document management functions such as locking, versioning, and editing.

1.2.6 Operations and deployment

WebSphere Portal Version 6 provides a number of enhancements in the area of operations and deployment. The enhancement are designed to give administrators more flexibility in the areas of security, configuration and availability.

Support for multiple LDAP realms

Previous versions of WebSphere Portal allowed for the use of a single LDAP directory for security and personalization information. In certain circumstances, primarily oriented around the use of Virtual Portals, this single LDAP directory domain could be partitioned into multiple subdomains through custom configuration.

This approach presented a number of problems both real and imagined, chiefly by exposing the single LDAP directory as a single point of failure and intrusion. These limitations could be mitigated through use of redundant LDAP servers, third-party network balancers, and complex recovery procedures designed to minimize downtime.

WebSphere Portal Version 6 enhances this capability in a number of key areas:

- Extends the concept of realms used primarily with *Virtual Portals* by allowing for multiple distinct LDAP directories, each tied to a specific Virtual Portal implementation.
- Provides for automatic failover to additional configured LDAP directory replicas without the need for an outboard load balancer.

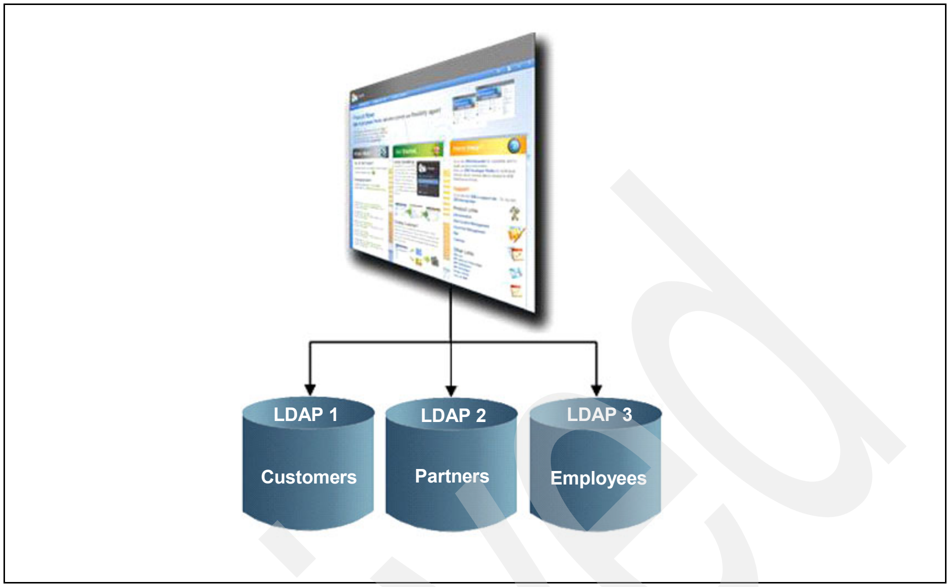


Figure 1-4 Multiple LDAP support

Database configuration enhancements

WebSphere Portal Version 6 provides for enhanced database configuration and management. The Portal configuration repository now consists of multiple, independently-managed databases and schema objects (database domains).

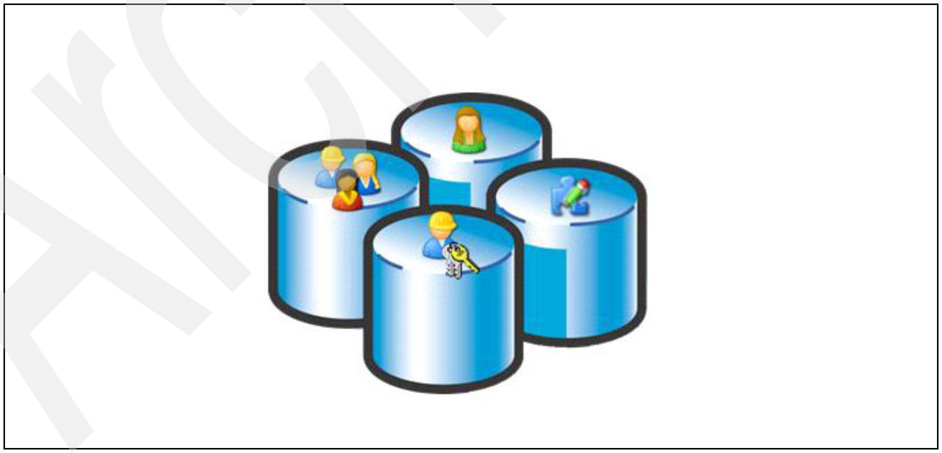


Figure 1-5 Database domains

The separation of Portal data into multiple domains provides more options for database administrators and system operators and administration personnel. By separating data, replication of the data potentially becomes easier and less expensive because not all types of data need to be replicated or backed up on the same schedule. Administrators may now create globally distributed Portal sites with regionally-specific data and domains, each with their own backup and availability requirements.

1.2.7 Enhanced application development and construction

Essentially, a *Portal* is a collection of individual applications presented in a common way with common formatting and rules. WebSphere Portal Version 6 provides new and innovative ways to build and integrate that content.

Simplified portlet creation

WebSphere Portlet Factory is a comprehensive portlet development environment that automates the process of creating, deploying, and maintaining Portals. Non-programmers can easily use WebSphere Portlet Factory to quickly create portlets that access existing data and transactions. These portlets can then be assembled into higher level functions that can be added to composite applications.

WebSphere Portlet Factory and WebSphere Portlet Factory Designer are not new products, but are now part of the WebSphere Portal offerings. The runtime and designer licenses are included in all versions of WebSphere Portal including IBM WebSphere Portal Enable for z/OS Version 6.0.0.1.

Workflow support

WebSphere Portal Version 6 includes a Workflow Builder tool as a technical preview. This tooling enables users to create and modify departmental workflows which, in a non-Portal environment, would have to be implemented by alternative means. (Note that this workflow is not supported in WebSphere Portal Enable for z/OS V6.0.0.1.)

Interactive forms in Portal interfaces

Business interactions today involve more than just reading and consuming information. Increasingly, interactions involve the use and completion of forms-based data. *Forms* are a pervasive way of exchanging data between businesses and consumers. WebSphere Portal integrates closely with functionality provided by the companion product, IBM Workplace Forms™. IBM Workplace Forms in conjunction with WebSphere Portal allows you to:

- Include electronic forms in a standard Portal interface.
- Enable users to easily access and exchange data with other applications.
- Allow users to collaborate to create, edit, and view electronic forms.

1.3 IBM WebSphere Portal Enable for z/OS Version 6.0.0.1

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 delivers a complete set of integrated, enterprise class Portal platform services. It exploits IBM System z and IBM z/OS in the areas of high availability, security, serviceability, scalability, proximity to data, parallel processing, and workload management.

IBM WebSphere Portal comes in three editions: WebSphere Portal Server, WebSphere Portal Enable, and WebSphere Portal Extend, as described here.

WebSphere Portal Server provides the core Portal infrastructure. WebSphere Portal Enable, and WebSphere Portal Extend add additional functionality in the areas of Web content and document management, workflow, instant messaging, and support for electronic forms. The version of Portal that ships for z/OS is WebSphere Portal Enable.

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 is available for the z/OS platform and runs on the WebSphere Application Server for z/OS V6.0.2. The IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 offering includes integrated support for Content Management, Document Management, and Personalization.

It enables collaboration with other IBM products such as Lotus Domino, Lotus Sametime®, and Lotus Quickplace, and it facilitates the deployment of portlets built with the IBM WebSphere Portlet Factory.

1.3.1 WebSphere Portal for z/OS features and functions

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 provides additional value above that provided for in the WebSphere Portal Version 6 for multiplatforms product. Those additional features and functions are outlined in the next section.

Support for the zFS file system

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 adds support for specifying zFS as a file system type. The HFS file system type has been stabilized by IBM and will not be enhanced.

zFS is the IBM strategic file system. It offers superior performance when compared to HFS and is the logical choice for creating shared file systems for use in the sysplex.

Refer to Example 1-1 on page 11 from the WebSphere Portal ISPF panel application.

Example 1-1   ISPF panel application file system configuration

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

 Configure a base portal into a base node
  System environment configuration

   Specify the following HFS information to configure your system
   environment. Then press ENTER to continue.

   Important:  Ensure you enlarge the WebSphere Application Server
   configuration HFS by the primary allocation you set here.


   WebSphere Portal HFS information

   Mount point....:  /waswpconfig/wpcell/wpport
   Name...........:  OMVS.WAS6.WPCELL.PORTALA.CONFIG.HFS
   Type (ZFS or HFS).................:  ZFS
   Volume, or '*' for SMS...........:  TST03F
   Primary allocation in cylinders...:  1600
   Secondary allocation in cylinders.:  50
```

Additional directory support

WebSphere Portal for z/OS now supports the standard set of directory choices enabled for all other IBM WebSphere based products. While not all configurations have undergone the same amount of testing as others, representative tests have been performed. Should you experience problems using a particular directory choice, contact IBM support for assistance.

Refer to Example 1-2 from the Portal ISPF panel application.

Example 1-2   ISPF panel application directory configuration

```
-----------------  WebSphere Portal for z/OS Customiz Specify valid LDAP type.
 Option  ===>

  Enable portal security using an LDAP registry without realms
   Basic LDAP settings (1 of 2)

    Specify the following to customize your WebSphere Portal for z/OS.
    Then press ENTER to continue.


    LDAP server type...................:
          ISO
```

```
       LDA
   EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e Correct values for LDAP server type are IBM (IBM SecureWay Security  e
LDA e Server for z/OS, or IBM Tivoli Directory Server), DOM (Domino e
e Directory), AD (Active Directory), ADA (Active Directory Application e
LDA e Mode), SUN (Sun ONE), and NDS (Novell eDirectory). e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
    LDAP server administrator password.:

    LDAP bind ID......................:
          wpsbind
    LDAP bind password................:


    Generate defaults for LDAP server type:  Y
```

Additional database connectivity options

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 supports the configuration of JDBC™ Type 4 database connectivity. Although it is primarily a configuration option for the database administrator, there are guidelines for choosing which option to use, as explained here.

JDBC Type 2

JDBC Type 2 is used to access a *local* DB2 subsystem resident in the same LPAR as the WebSphere Portal. It is a high speed native connector built on z/OS-specific technology (RRSAF). These are the benefits of using a JDBC Type 2 connector:

- It provides support of full two-Phase Commit (2 PC) processing using z/OS Resource Recovery Services (RRS).
- It is more efficient than 2-PC processing provided by J2EE™ XA.
- It provides support of DB2 data sharing in a sysplex environment.
- The application's SQL work takes place within the same enclave as the Java work, which means that WLM can manage resources across both WebSphere and DB2 to ensure that the WebSphere workloads target is met.

JDBC Type 4

JDBC Type 4 is used to access a *remote* DB2 subsystem resident in another LPAR or another physical z/OS system. It is based on network-based protocols (TCP/IP and DB2 Distributed Data Facility (DDF)). The advantage of using JDBC Type 4 is flexibility, which means having the ability to implement the database on another LPAR or server than the WebSphere Portal.

However, there are several disadvantages in using JDBC Type 4:

- There is more overhead for global transactions (2-PC provided by J2EE XA).
- There is more overhead for security and parameter marshalling and demarshalling.
- There is more overhead because of a “pure” Java implementation.
- It does not support DB2 data sharing in a sysplex environment.
- An application's workload is a separate enclave from the Java part. WLM sees the Java and SQL as completely separate workloads, and does not understand that they are related.

  Therefore, when the system is under stress, WLM does not know to increase DB2 priority, for example, in order to ensure that a WebSphere response time target is met.

JDBC Type 4 may be attractive for customers with an isolated zIIP environment.

Refer to Example 1-3 from the Portal ISPF panel application.

Example 1-3 ISPF panel application JDBC driver configuration

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

  Transfer database
   Database driver configuration (2 of 2)

    Specify the following for the system on which you are installing
    WebSphere Portal. Then press ENTER to continue.


    DB2 location name....................:  LOC1
    DB2 subsystem name...................:  DB2
    JDBC driver type (2 or 4)............:  2

    DB2 JCC properties file (type 2 only):
       /etc/DB2JccConfiguration.properties

    DB2 server name (type 4 only)........:
       yourdb2server.com
    DB2 port number (type 4 only)........:
       446
```

Improved database creation jobs and scripts

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 allows z/OS database administrators to tailor the configuration of their DB2 for z/OS installations.

Because WebSphere Portal now requires up to seven different database definitions, this ability is extremely important. The parameters that can be tailored include:

- Schema name
- Storage group
- VCAT
- Volumes
- Buffer pool names

Refer to Example 1-4 from the WebSphere Portal ISPF panel application.

Example 1-4   ISPF panel application database configuration

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

  Transfer database
   Database configuration  (1 of 7)

    Specify the following for the system on which you are installing
    WebSphere Portal. Then press ENTER to continue.


    Database configuration for domain Release

      Database name.............:  WPSDBREL
      Database schema name......:  RELEASE
      Database user ID..........:  WPSDBADM
      Database password.........:  WPS2BADM

      Database storage group....:  WPSSG
      Database volumes..........:
          *
      Database VCAT.............:  DSN810
      Database 4K buffer pool...:  BP0
      Database 32K buffer pool..:  BP32K
      Database index buffer pool:  BP0
```

1.4 WebSphere Portal prerequisites and corequisites

The IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 product is a rich set of features and capabilities. It relies on many cooperating products to provide this value. This section of our work details the hardware and software prerequisites and corequisites necessary for a successful Portal for z/OS deployment.

1.4.1 Hardware prerequisites

The following hardware items are required for deploying WebSphere Portal on z/OS:

- Processor - any current System z class processor that is capable of running IBM WebSphere Application Server V6.0.2.17.
- Physical memory - the *minimum* real memory requirement is 2 GB per z/OS logical partition (LPAR). On our test systems we used 6 GB per LPAR.
- Disk space
  - Install file system - 1.5 GB or 2300 cyls
  - Configuration file system - 1.5 GB or 2300 cyls per unique server instance
  - /tmp directory space - at least 300 MB
    - Also, increase the primary space allocation of the WebSphere Application Server for z/OS /Config directory by the same amount (at least 1600 cyls).
  - Ensure there is enough free space in directories WAS_HOME/temp, and /WAS_HOME/wstemp, preferably about 300 MB in each directory.
  - After allocating or enlarging HFS data sets, check that your SMS rules have not given a lower primary space allocation than you requested.
  - Approximately 9GB of disk space is required for an instance of the Portal DB2 databases after you have transferred the Portal database to DB2.

In addition to these hardware requirements, the use of one or more System z Application Assist Processors (zAAPs) is highly recommended.

1.4.2 Software prerequisites

There are a number of required software prerequisites for deploying WebSphere Portal on z/OS. The most current list can be found in the supported hardware and software for WebSphere Portal V6.0 section at the Infocenter site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/inst_req60.html\

For completeness, we list the current prerequisites and product levels here:

- z/OS V1.7, or V1.8
- z/OS V1.6 - including PTFs UA17067, UA17068, and UA17069
- RACF®, ACF/2, or Top Secret at a level compatible with the deployed operating system level

- SMPE for z/OS V3.0.2
- Java SDK 1.4.2 - Level SR1A UK00802 VM build cm142sr1aifx-20050310 (as reported at a UNIX shell prompt by the java -fullversion command) and the relevant PTF for APAR PK01572
- UNIX ported tools: zip, unzip (the user ID installing the WebSphere Portal product using SMPE, and any user ID configuring a WebSphere Portal, must have the zip/unzip tools available in its UNIX PATH)
- IBM WebSphere Application Server for z/OS V6.0.2.18

> **Note:** Using WebSphere Application Server for z/OS V6.0.2.17 is not recommended because it does not support clustering. Clustering only works when using WebSphere Application Server V6.0.2.19, and when applying a fix on WebSphere Portal that was not available at the time of writing.

- Web browser
  - Microsoft® Internet Explorer® 6.0 SP1 or SP2 (required for Win XP)
  - Firefox 1.5.0.3
  - Mozilla Web Browser 1.7
  - Netscape Communicator 8.1
  - Apple Safari 2.0

1.4.3 Optional software prerequisites

There are many companion products that may be installed in support of WebSphere Portal on z/OS. The products and versions listed in the following section integrate directly with WebSphere Portal on z/OS through one or more WebSphere Portal-supplied configuration tasks.

In addition, there are an almost infinite number of ways to integrate functions into the WebSphere Portal through deployed Portlets and supporting server functions, but it is beyond the scope of this book to list all those possible products and functions.

The most current list of optional software corequisites can be found in the supported hardware and software section of the IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 Infocenter site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/inst_req60.html

For completeness, we list the most commonly used optional prerequisites here.

Web server

- Apache Server 2.0.49 or 2.0.52
- IBM HTTP Server 2.0.47.1, 6.0, 6.0.1, or 6.0.2
- IBM HTTP Server for z/OS
- IBM HTTP Server for iSeries® (powered by Apache)
- Internet Information Server 5.0 or 6.0
- Lotus Domino Enterprise Server 6.0.3, 6.0.5, 6.52, or 6.5.4
- Sun ONE WebServer 6.0 SP7 or SP9
- Sun Java™ System Web Server 6.1 SP1 or SP3

Database

- Cloudscape™ 5.1.60.38 - automatically installed during Portal for z/OS installation
- DB2 UDB for z/OS V7.1 or V8.1

  PUT level 0602 and PTF PK25139

LDAP server

- IBM Secureway Security Server for z/OS V1.6, V1.7, or V1.8
- IBM Tivoli® Directory Server 5.2 or 6.0
- IBM Tivoli Directory Server for z/OS 1.8
- IBM Lotus Domino 6.5.4, 6.5.5, or 7.0.1
- Novell eDirectory 8.7.3
- Sun Java System Directory Server 5.2
- Microsoft Active Directory® 2000 or 2003
- Microsoft Active Directory Application Mode (ADAM) 2003

Collaboration

- IBM Lotus Domino 6.0.2, 6.0.3, 6.0.4, 6.5.1, 6.5.2, 6.5.3, 6.5.4, 6.5.5, or 7.0.1
- IBM Lotus Sametime 7.0 or 7.5
- IBM Lotus Instant Messaging and Web Conferencing 6.5.1
- IBM Lotus Quickplace 7.0
- IBM Lotus Team Workplace 6.5.1

Content management

- IBM Workplace Web Content Management 6.0 - automatically installed during Portal for z/OS installation
- Document Manager - automatically installed during Portal for z/OS installation
- Personalization - automatically installed during Portal for z/OS installation

Security

- IBM Tivoli Access Manager for e-business 5.1 or 6.0

Portlet development

- Rational Application Developer 7.0
- WebSphere Portal Application Integrator Development Tool 5.1
- IBM WebSphere Portlet Factory 6.0

1.5 WebSphere Portal and SOA

Service Oriented Architecture (SOA) is not a product or a packaged solution. It is a set of architectural principles that examine the relationships between business functions and processes, and how these primary building blocks can be assembled easily and effectively in new and interesting ways.

SOA enables business flexibility by enabling the creation of new business processes using common, well-understood techniques that respond well to change. Business process and composite applications can be built from new and existing components by focusing on what is differentiating, not by strictly examining requirements. A well-planned SOA encourages the reuse of existing assets, while providing flexibility in implementing new functions.

As this definition suggests, the most important word in Service Oriented Architecture is “architecture”. SOA is in many ways an extension of well-defined architectural methods and principles. What SOA adds is a set of structured frameworks and a methodology for implementation that lends itself to today’s highly Web-centric world.

1.5.1 The SOA Lifecycle

The *SOA Lifecycle*, as defined by IBM, involves four phases:

- Model - use modeling tools to create a business-oriented model of the business process that will later be used to generate code for the assembly phase.
- Assemble - build or develop the application using the proper collection of orchestrated services. Link the services together using an Enterprise Service Bus to provide an abstraction layer that eliminates the need for tight coupling of services.
- Deploy - place the composite applications onto runtime systems that are appropriate for the qualities of service needed for the application.
- Manage - monitor and control the components of the applications using tools that monitor not only IT execution characteristics (performance, response time, and so on), but also business results.

Figure 1-6 illustrates these phases.

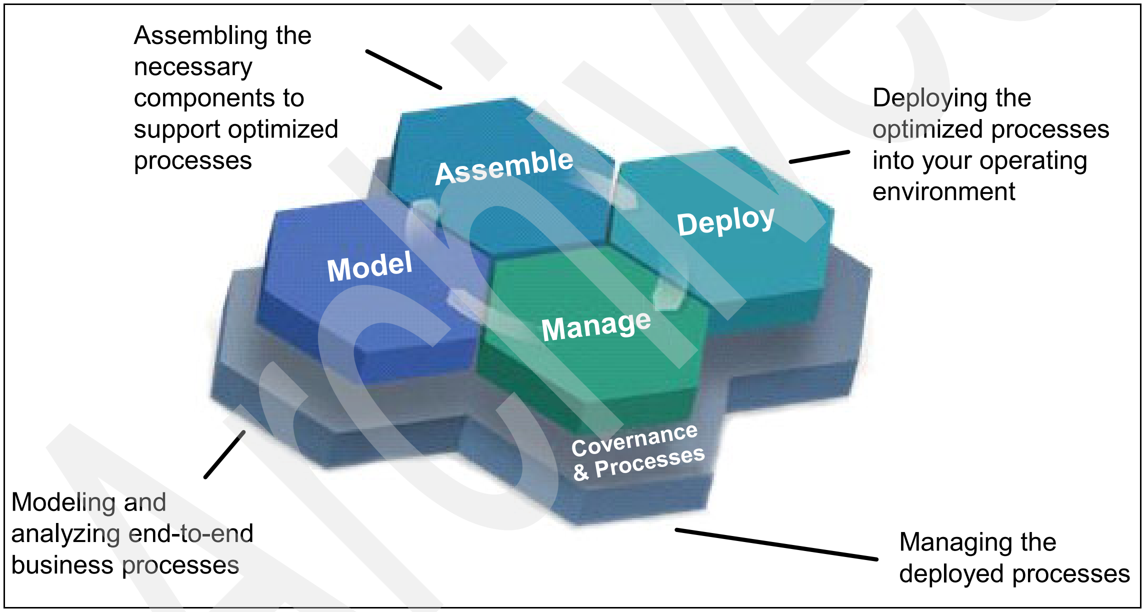

Assembling the necessary components to support optimized processes
Deploying the optimized processes into your operating environment
Assemble
Deploy
Model
Manage
Covernance & Processes
Modeling and analyzing end-to-end business processes
Managing the deployed processes

Figure 1-6 The SOA Lifecycle

1.5.2 The SOA Reference Architecture

The *SOA Reference Architecture* is a way of looking at the set of services that go into building an SOA. These capabilities can be implemented as needed, thus allowing functions and project level solutions to be easily added as new requirements are surfaced over time.

In Figure 1-7, the SOA Reference Architecture model groups functions in support of the SOA lifecycle. Components on the far left support Model and Assemble. The components on the far right represent management of the SOA. The components in the center illustrate the deployment phase of the SOA Lifecycle.

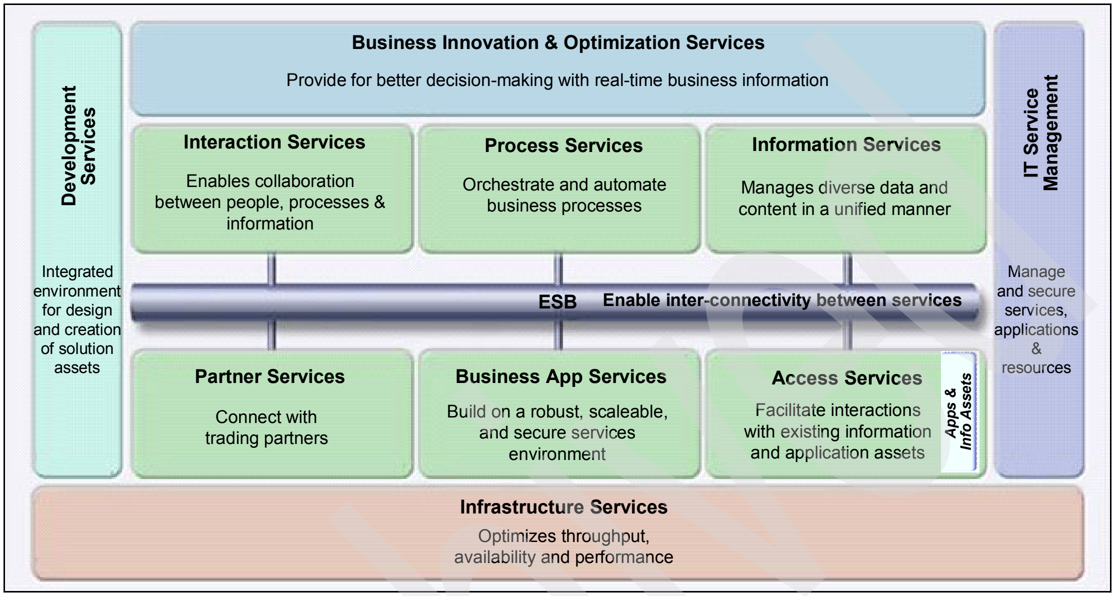

Development Services
Integrated environment for design and creation of solution assets
Business Innovation & Optimization Services
Provide for better decision-making with real-time business information
Interaction Services
Enables collaboration between people, processes & information
Process Services
Orchestrate and automate business processes
Information Services
Manages diverse data and content in a unified manner
ESB
Enable inter-connectivity between services
Partner Services
Connect with trading partners
Business App Services
Build on a robust, scaleable, and secure services environment
Access Services
Facilitate interactions with existing information and application assets
Apps & Info Assets
IT Service Management
Manage and secure services, applications & resources
Infrastructure Services
Optimizes throughput, availability and performance

Figure 1-7 The SOA Reference Architecture

1.5.3 WebSphere Portal is the first major SOA application

"Through 2007, an enterprise portal will be the first major application of SOA concepts for more than 50 percent of enterprises."

Gene Phifer, Gartner Research, "Management Update: A Portal May Be Your First Step to Leverage SOA", October 12, 2005.

A people--centric approach to SOA focuses on the end-user experience to facilitate innovation and greater collaboration. People-centric collaboration is about human and process interaction with predictable levels of service.

SOA enables the user experience and ensures consistent service levels each step of the way. SOA allows customers to create, deploy, and update composite applications faster by allowing developers to focus on that part of the application infrastructure where innovation and invention is most needed.

Portals allow the presentation space to be partitioned into smaller pieces called *portlets*. As applications are created or converted to service-based portlets, portal administrators can aggregate this new and enriched content faster and more economically to get a more collaborative user interface.

1.5.4 Portal is the front end of SOA

Today, applications and information are delivered in silos. Portal allows for an integrated environment where the user experience drives interaction with new and existing infrastructure to enable robust, composite application construction using SOA principles and architecture; see Figure 1-8.

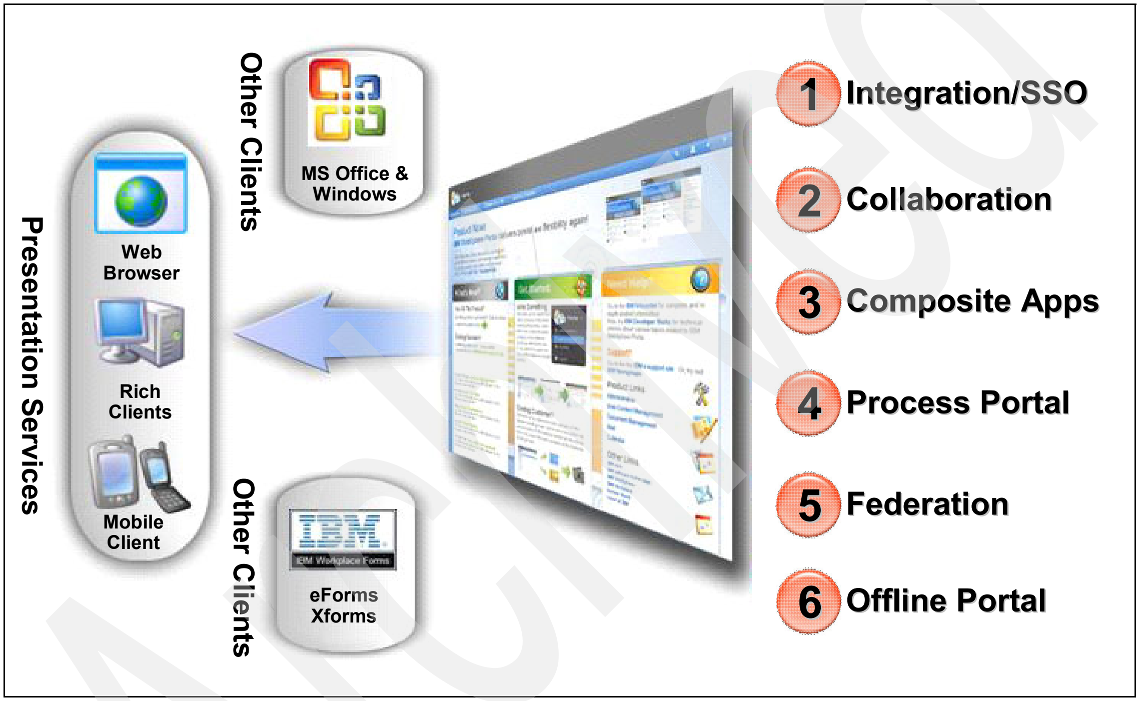


Figure 1-8 Portal is the user experience for SOA

1.5.5 The power of SOA with WebSphere Portal

Services can be built with a variety of tools and developer skills; see Figure 1-9 on page 22. Services can be built independently from the UI assembly.

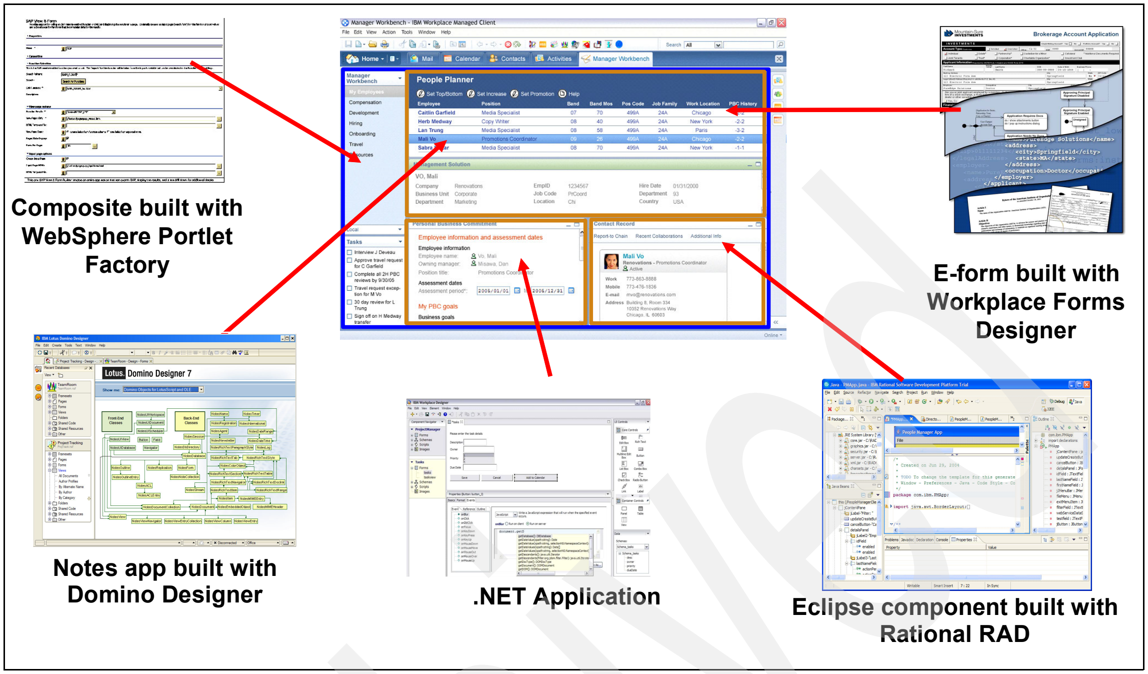

Composite built with WebSphere Portlet Factory
E-form built with Workplace Forms Designer
Notes app built with Domino Designer
.NET Application
Eclipse component built with Rational RAD
People Planner
Manager Workbench - IBM Workplace Managed Client
Lotus Domino Designer 7

Figure 1-9 Building services for Portal deployment

1.6 Common WebSphere Portal architectures

WebSphere Portal for z/OS may be configured in a variety of ways, from a very basic setup using a single z/OS LPAR, a standalone WebSphere Application Server, a Cloudscape database and no security to a fully configured HA failover environment with multiple WebSphere-managed nodes, LDAP or external security, and multiple servants. In the following section we explain some common configurations.

1.6.1 WebSphere Portal in a standalone WebSphere Application Server

This most basic configuration, outlined in Figure 1-10 on page 23, illustrates a single instance of WebSphere Portal configured in a standalone WebSphere Application Server. Notice that the WebSphere Application Server on z/OS consists of multiple physical regions: a *Control region* and one or more *servant regions*.

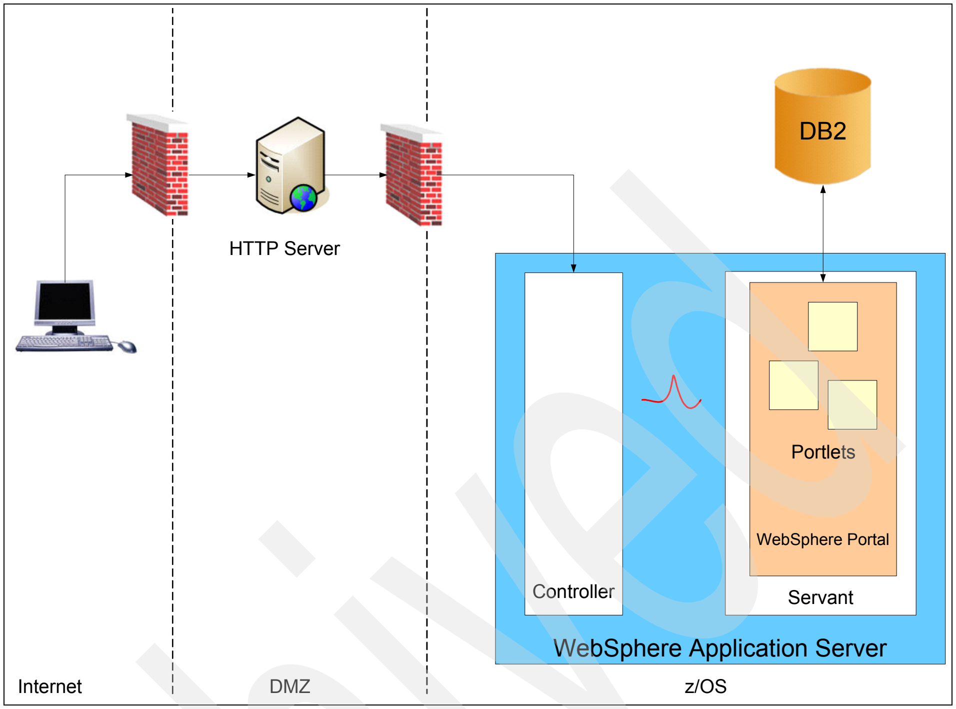


Figure 1-10 Portal z/OS base node configuration

The Control region receives inbound HTTP requests from the network via an HTTP server located in a *demilitarized zone* (DMZ) protected by firewalls. The HTTP server is set up as a reverse proxy, accepting inbound requests from the network, but allowing no traffic to flow directly to the protected z/OS zone.

The Control region passes work requests over one or more work queues to Servant regions. The Servant regions contain a full copy of WebSphere Application Server, WebSphere Portal and a JVM™. Within the JVM is a full Java execution environment for all WebSphere Portal artifacts.

The Portal constructs HTTP pages for rendering on the client browser. It passes this HTML back to the Control region, which in turn returns the data via the HTTP proxy to the client browser.

We show the WebSphere Portal using a DB2 database for configuration data. This indicates that the database transfer operation has been performed for the Portal. There is no Portal security configured in this scenario.

Configuring WebSphere Portal in a WebSphere standalone server is very straightforward and is therefore a good starting point for learning the Portal environment. It does, however, have some serious limitations. The biggest limitation is that Portal instances configured into a standalone WebSphere Application Server may *not* be migrated to a managed node configuration. Be sure to keep this fact in mind when building a Portal configuration in a WebSphere standalone server.

Also, note that a Cloudscape database does not support transactional integrity when there are multiple servants. So when using Cloudscape, the Portal in the WebSphere Application Server standalone server is limited to one servant.

1.6.2 WebSphere Portal in a managed node of a WebSphere Application Server Network Deployment configuration

For the scenario illustrated in Figure 1-11 on page 25, we show a Portal configuration identical to the previous configuration. The only difference is in the WebSphere Application Server configuration. In this case, we have configured the Portal in a WebSphere-managed node. The managed node configuration adds two major components.

The WebSphere Deployment manager communicates with a browser-based administrative client and pushes information to one or more Node Agents. The Node Agents in turn communicate with any servant regions resident in their respective nodes. The Node Agents are responsible for communicating any administrative changes to their servers, as well as any z/OS server commands such as Stop, Start, and Modify commands.

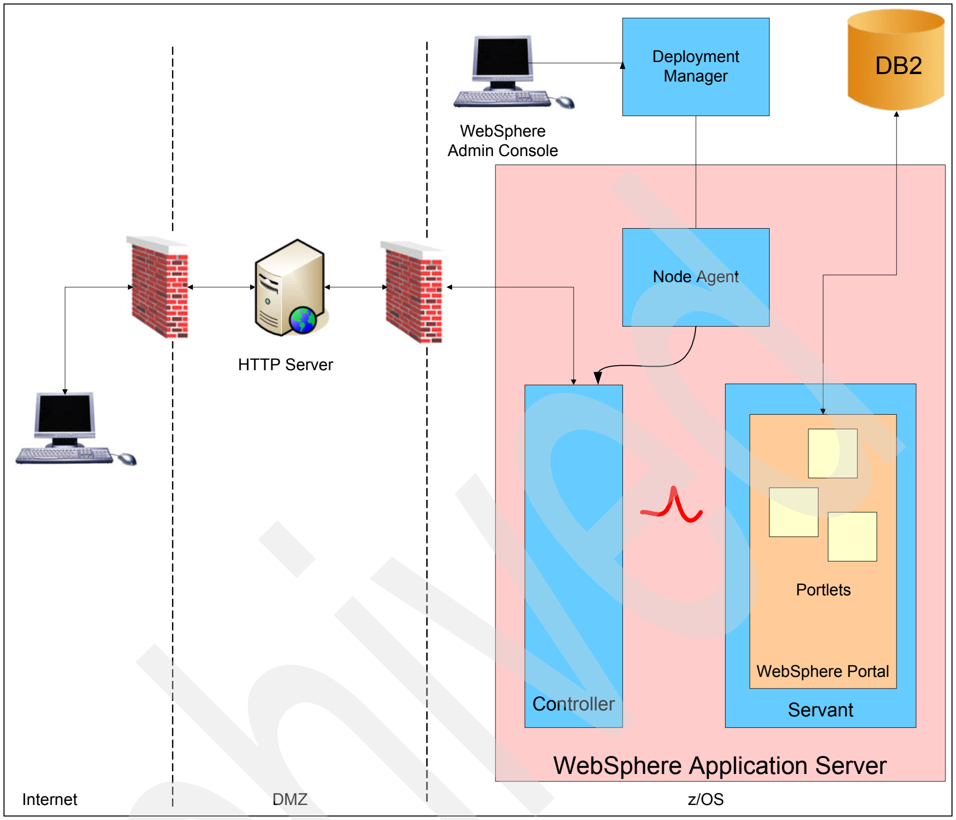


Figure 1-11 Portal z/OS-managed node configuration

The major advantage of a WebSphere-managed node over a standalone server is in the ease of deploying changes across a set of servers. By communicating with the Deployment Manager, an administrator can make a single change in one place for a set of Portal servers. Contrast this with the many separate commands and tasks that would be necessary if communication were required with each and every Portal server.

1.6.3 WebSphere Portal with LDAP security enabled

No WebSphere Portal installation is complete without configuring security. There are two major classes of security solutions for use with WebSphere Portal. The scenario illustrated in Figure 1-12 on page 26 highlights the first security solution.

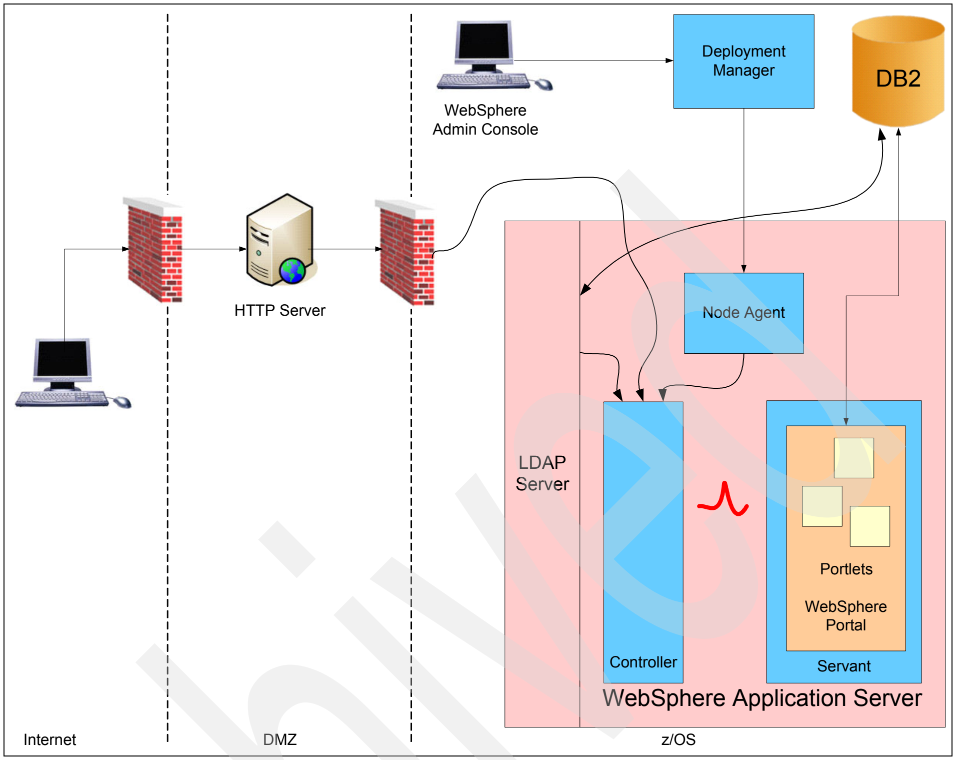


Figure 1-12 WebSphere Portal for z/OS-managed node with LDAP security

WebSphere Application Server for z/OS provides three user registry types for use in authentication exercises: LDAP, LocalOS, and custom. WebSphere Portal for z/OS supports two types of user registries: LDAP and Database. It relies on the underlying WebSphere Application Server for z/OS LDAP or Custom user registry capabilities.

The simplest form of LDAP-based authentication is outlined here. In our scenario we used the IBM Secureway LDAP server for z/OS. When a protected page is encountered, WebSphere passes all authentication requests to the WebSphere Portal login module.

WebSphere Portal uses WebSphere-based security services and the configured LDAP directory to create a user credential for the request. This credential flows with the request through all WebSphere Portal processing. In addition, depending on the security options configured for base WebSphere, this credential can be used to access back-end host-based systems.

1.6.4 WebSphere Portal with external security enabled

Portal can also be configured to use an external authentication service such as IBM Tivoli Access Manager. In the scenario illustrated in Figure 1-13, we show a simplified topology with a Tivoli Access Manager proxy server in the DMZ.

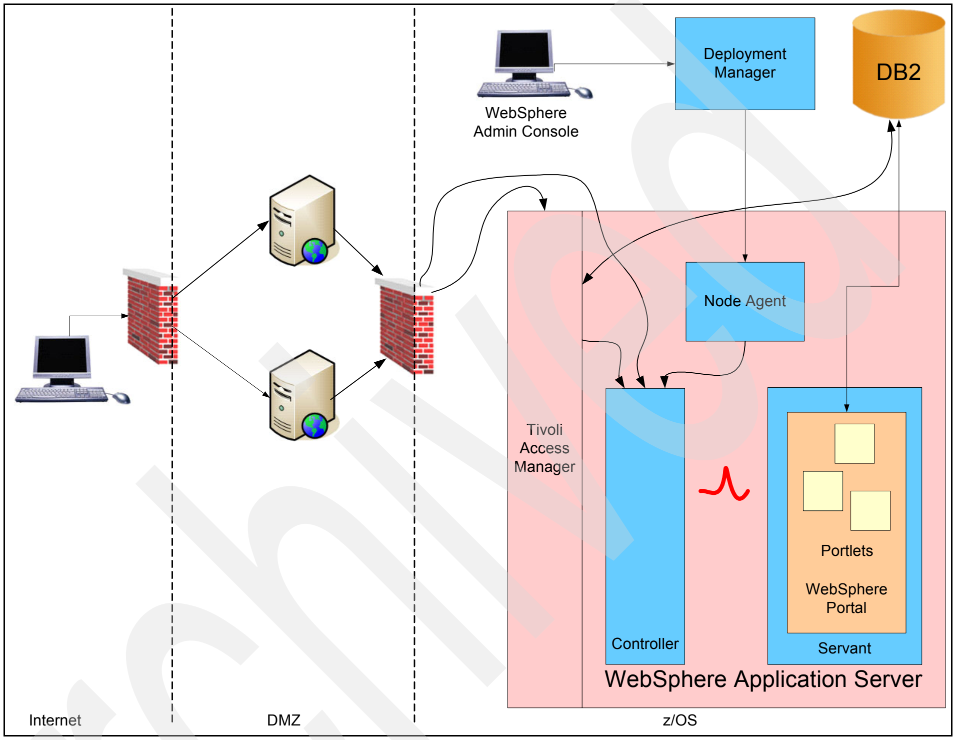


Figure 1-13 WebSphere Portal for z/OS-managed node with external security

WebSeal authenticates user requests and then, depending on the configuration, either creates and passes a Lightweight Third Party Authentication (LTPA) token with the request or an identity in an HTTP header which is read by a Trust Association Interceptor (TAI) on the WebSphere side. In either case, WebSphere creates a user credential based on the identity of the user authenticated by the WebSeal proxy server.

1.6.5 Portal and vertical scaling

WebSphere Portal for z/OS does not support vertical scaling by defining multiple nodes on a single logical partition (LPAR) as you might do with WebSphere on non-z/OS platforms. That is, although it is possible to physically configure a cell with more than one node on an LPAR, and with a Portal server instance in each node, for Portal on z/OS this turns out to be a waste of system resources.

On z/OS, vertical scaling is achieved by enabling more than one servant region per control region. Because servant regions are managed by the z/OS operating system and the *Workload Manager* (WLM), a single control region can easily route work requests to multiple servant regions without the extra overhead of another control region. WLM can start additional servant regions as necessary to fully utilize processor and memory capacity.

The Portal configuration for the scenario illustrated in Figure 1-14 is exactly the same as for the single servant region configuration. Additional minor configuration is required in the application server to enable these features.

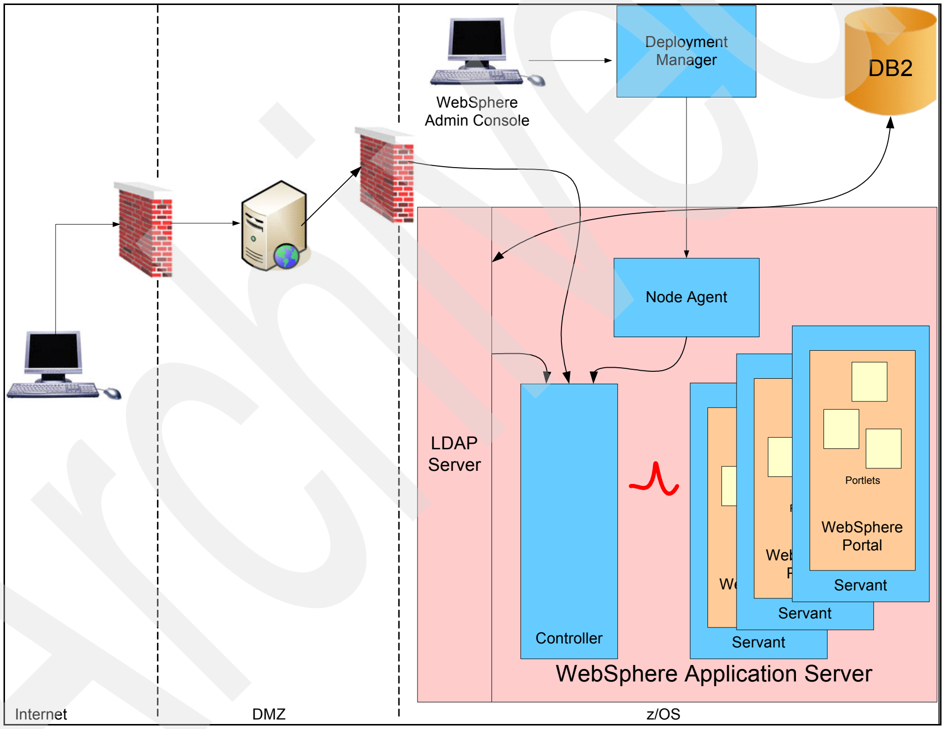


Figure 1-14 WebSphere Portal for z/OS-managed node with vertical scaling

1.6.6 Portal and z/OS Parallel Sysplex

The Portal configuration illustrated in Figure 1-15 on page 29 consists of a cell with two nodes. The LPARs in the Parallel Sysplex® are connected via a Coupling Facility (CF), and a DB2 database is configured in *datasharing* mode.

There is one node in each LPAR of a two-LPAR sysplex. The Portal is configured in a cluster, and work is distributed between the Portals running in the two nodes.

To this configuration we have added an HTTP Edge server and an IP sprayer. The IP sprayer is configured to accept inbound HTTP requests for the sysplex. It passes these requests downstream to the HTTP servers. The HTTP servers are configured to distribute requests to both systems in the sysplex.

To simplify the diagram we have omitted all firewalls and proxies. Note this is one of many configurations possible with WebSphere Portal and the z/OS sysplex.

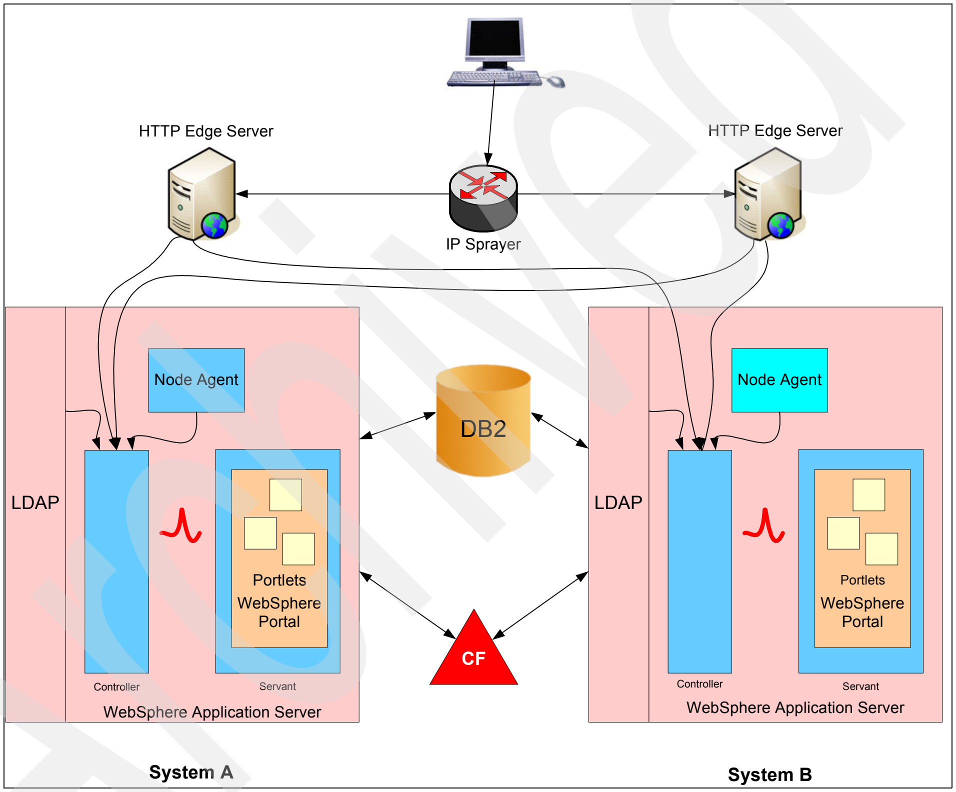


Figure 1-15 WebSphere Portal for z/OS sysplex with horizontal scaling

Note that LDAP also benefits from the sysplex capabilities. In this configuration the LDAP servers on both LPARs can share a single copy of the LDAP TDBM database in DB2. If any LPAR is down or the LDAP server on any of the LPARs is down, the LDAP servers on the other active LPARs still have access to the same LDAP database. This way, Portals installed on different LPARs can use the same security data. This LDAP High Availability solution is provided “out of the box” on z/OS.

1.7 The value of IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 and System z

IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 exploits System z hardware and the z/OS operating system to provide a robust computing platform. Some of the ways WebSphere Portal exploits System z and z/OS qualities of service are described in the following sections.

1.7.1 64-bit operating system designed for mainframe computing

The latest System z hardware and the z/OS operating systems support 64-bit computing. An increasing number of software solutions running in z/OS are becoming 64-bit enabled, as well. Running in 64 bit provides a number of advantages:

- Serving an increasing number of concurrent users
- Running larger workloads
- Running mission-critical applications even more securely
- Managing an increasing number of background (batch) jobs
- Controlling an increasing number of I/O devices

Important: Note that IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 itself is not 64-bit enabled yet, because it runs on top of WebSphere Application Server V6.02, which is not 64-bit enabled yet.

However, IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 can benefit from 64 bit now by using z/OS functions and other subsystems (such as DB2) that are already 64-bit enabled. At the time of writing we expect that future versions of WebSphere Portal for z/OS will be 64-bit enabled when they run on top of WebSphere Application Server V6.1.

1.7.2 Server-based workload management

z/OS provides true server-based workload management. It samples all key workload parameters for all work running in the system. It collects these samples four times a second into rolling histories. Using this data and the installation WLM policy in effect, the WLM code makes policy decisions every ten seconds about when and which system resources to adjust to accomplish the goals set forth by the installation. WLM and z/OS require no outside intervention to make these adjustments.

1.7.3 Based on WebSphere Application Server

WebSphere Portal for z/OS Version 6 runs on top of WebSphere Application Server for z/OS, and exploits its capabilities. WebSphere Application Server running on z/OS provides the following benefits to WebSphere Portal for z/OS:

- WebSphere Application Server for z/OS is integrated with the z/OS operating system at all levels. Through its deep integration, it is able to exploit the superior qualities of service the platform provides. Because WebSphere Portal for z/OS runs within this environment, all of these benefits are extended to WebSphere Portal for z/OS.
- WebSphere Application Server for z/OS is implemented as multiple, distinct processes. The controller process accepts incoming connection requests (HTTP, IIOP, Messages) and forwards them to one or more servant processes via a set of queues managed by the z/OS Workload Manager (WLM).
- The controller is a system process and runs authorized, key 0. No application code may run in this process, and it is therefore isolated from any “bad” behavior an application may exhibit.
- The servants are system processes started by the controller with WLM’s help. Servants run problem state, key 8 application code and may not touch any resources outside their execution environment.

  Each servant is an isolated Java execution environment with its own resources and its own requirements. Through the installation WLM policy different servants may process different work with different characteristics including transaction, security, performance, and accounting.
- The number of servants can be managed dynamically by WLM, which can increase or decrease the number of servants in response to changing workloads. In the event of an application failure that takes a servant down, WLM will initiate a restart of another servant.

  Even if you have only one, non-clustered Portal, with Portal on z/OS you can have multiple servants which gives you protection against the most common application problems that cause a servant JVM failure.

1.7.4 Proximity to data

z/OS provides optimized connectors for accessing mainframe-based resources such as DB2 data and CICS and IMS™ transactions. Through these optimized connections, WebSphere Portal can achieve superior response times and transaction throughput rates when accessing mainframe data and transactions.

1.7.5 Mixed workloads

z/OS is optimized for running multiple workloads concurrently using a common set of hardware resources. This is achieved through superior virtualization techniques employed at multiple places in the hardware and software stacks. From a WebSphere Portal standpoint, this means higher CPU utilization can be achieved when running WebSphere Portal on z/OS.

1.7.6 Optimized network attachments

The z/OS Communications Server is optimized for handling multiple network protocol traffic over multiple network attachments in a flexible and reliable configuration. WebSphere Portal exploits these network qualities of service to provide superior network bandwidth and availability to its end users.

1.7.7 Integrated security design

z/OS works together with the System z hardware to provide a secure computing platform with multiple recovery levels built in. z/OS works together with the Resource Action Control Facility (RACF), or equivalent vendor products, to protect and secure mainframe resources including those resources used by the WebSphere Application Server and WebSphere Portal.

1.7.8 Specialized processors

The System z family of hardware processors provide a highly competitive Java processing environment. This environment is created by employing one of more Java co-processors. These co-processors or System z Application Assist Processors (zAAPs) provide a very cost-effective way of running Java workloads on z/OS.

1.7.9 Hardware exploitation and acceleration

IBM has invested in key hardware technologies to better leverage the capabilities of the hardware in performing common software functions faster and more efficiently. Some of these enhancements include: IEEE floating point, checksum, dedicated lock operations, embedded SSL, and integrated crypto facilities.

1.7.10 Parallel Sysplex abilities

Parallel Sysplex allows you to harness the power of up to 32 separate z/OS systems. These systems, while separate, provide a single system view of compute power and resources to networks, users, and most importantly, applications.

To accomplish this goal, Parallel Sysplex employs two unique abilities:

- It creates an environment for true peer-to-peer parallel processing among the systems that make up the sysplex.
- It enables read/write data sharing among members of the sysplex through the use of a hardware facility called a Coupling Facility (CF).

These two technologies, when combined with applications like WebSphere Application Server that exploit them, create a solution that cannot be matched by any other system, solution, or architecture today.

1.7.11 Operator controlled and monitored

The z/OS operating system and all of its constituent parts, including all of the applications that run on top of it, is fully controlled and monitored via multiple operator interfaces. System and application operation may be automated to whatever extent is required by an installation. Multiple entry points into the system infrastructure provide robust logging, tracing, and audit capabilities.

2

WebSphere Portal for z/OS primary node configuration

In this chapter we describe the WebSphere Application Server for z/OS environment that we used for the installation of IBM WebSphere Portal Enable for z/OS Version 6.0.0.1.

We also explain how to configure WebSphere Portal for z/OS into a federated node within a WebSphere Application Server for z/OS cell.

2.1 Initial software configuration

In this scenario, we used the following software:

- z/OS Release 1.8
- DB2 Version 8 (datastore for the relational databases) used for:
  - Portal databases
  - LDAP TDBM databases
- RACF (infrastructure security)

  RACF contains the user profiles for the WebSphere Application Server and several authorizations required to access the server privileges
- WebSphere Application Server Version 6.0.2.18
- WebSphere Portal Version 6.0.0.1 (extension to WebSphere Application Server)

2.2 Setting up WebSphere Application Server

> **Tip:** It is beyond the scope of this book to discuss all the best practices related to defining a WebSphere z/OS Network Deployment configuration. We strongly recommend that you obtain white paper WP100653 from the following site, and base your naming conventions and best practices on the information provided by that paper.
>
> `http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100653`

For the initial WebSphere Application Server setup in our scenario, we installed a WebSphere Network Deployment (ND) configuration with an empty node. WebSphere Portal can also be configured on a base application server with a federated empty node.

We chose a Network Deployment installation because in later chapters we discuss using WebSphere Portal in a high availability, clustered environment with multiple nodes and application server clones, and a Network Deployment configuration is required for clustering.

The WebSphere z/OS cell that we configured in order to host the Portal is called WPCELL.

This cell contains the following:

- A daemon called WPDEMNA.
- A Deployment Manager server composed of control region WPDMGR and servant region WPDMGRS, located in the Deployment Manager node.
- An empty managed node WPNODEA with its Node Agent WPAGNTA. It is in this node that we installed the WebSphere Portal.

Figure 2-1 illustrates the initial WebSphere Application Server setup that we used.

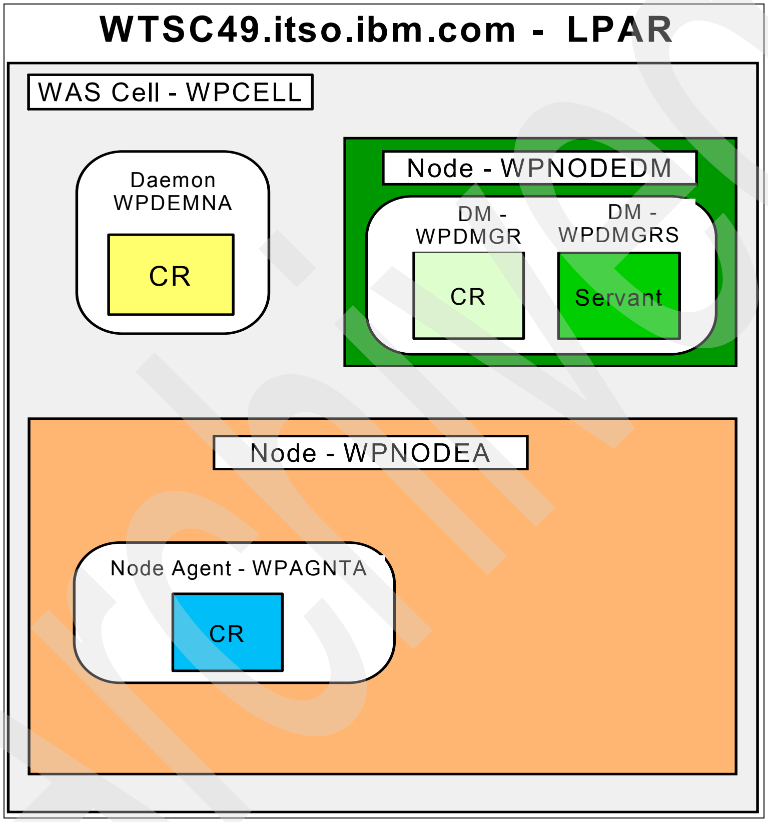
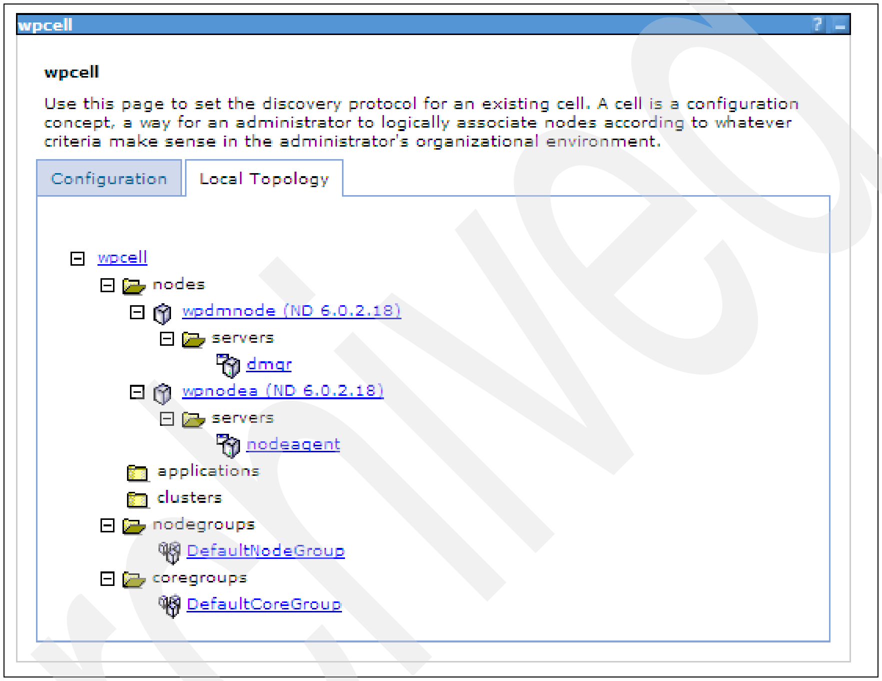


Figure 2-1 WebSphere Application Server setup before installing WebSphere Portal

At this point, an application server had not yet been defined for our WebSphere Application Server Network Deployment installation. This WebSphere Portal installation was done into an empty, managed node with no other application servers. One of the Portal configuration jobs, EJPSNDC1, runs JACL scripts that, among other things, will create an application server to host the Portal.

Figure 2-2 illustrates our initial topology as seen via the WebSphere Admin console. The path to it is **System Administration -> Cell -> Local Topology**.



wpcell

Use this page to set the discovery protocol for an existing cell. A cell is a configuration concept, a way for an administrator to logically associate nodes according to whatever criteria make sense in the administrator's organizational environment.

- wpcell
  - nodes
    - wpdmnode (ND 6.0.2.18)
      - servers
        - dmgr
    - wpnodea (ND 6.0.2.18)
      - servers
        - nodeagent
  - applications
  - clusters
  - nodegroups
    - DefaultNodeGroup
  - coregroups
    - DefaultCoreGroup

Figure 2-2 WebSphere Application Server initial topology

Throughout this book we use names of data sets, HFSs, and so on that relate to the values that we used during our installation process. Table 2-1 lists all of these values in node 1. You can use this table as a reference for many of the examples that are presented throughout the book.

Table 2-1 Values used in our installation process

Parameter	Value
WebSphere Application Server datasets	`BBWP6049.SBBO* (prefix)`
WebSphere Portal Server datasets	`BBWP6049.SEJP* (prefix)`
WebSphere Application Server SMP HFS	`/usr/lpp/zWebSphereWP/`
Portal SMP HFS	`/usr/lpp/zPortalServerWP`
Portal root HFS/dataset	`/waswpconfig/wpcell/wpport/`
	`OMVS.WAS6.WPCELL.PORTALA.CONFIG.HFS`
Application Server Root HFS/dataset	`/waswpconfig/wpcell/wpnode/AppServer/`
	`OMVS.WAS6.WPCELL.WPNODEA.CONFIG.HFS`
Deployment Manager Root HFS/dataset	`/waswpconfig/wpcell/wpdmnode/DeploymentManager/`
	`OMVS.WAS6.WPCELL.WPDMNODE.CONFIG.HFS`

2.2.1 WebSphere Application Server custom settings

You must make several configuration changes to the WebSphere Application Server ND setup before you can install WebSphere Portal. Here is a list describing each change:

1. Increase the size of the HFS defined for the Deployment Manager to the size of the HFS that will be used for WebSphere Portal.

   WebSphere Portal has a minimum requirement of 1600 cylinders for its HFS, so the Deployment Manager HFS needs to have at least 1600 cylinders.

   – The Portal .ear and .war files will be placed in the configuration HFS data sets of both WebSphere Application Server for z/OS and WebSphere Portal for z/OS during configuration. Enlarging the HFS ensures that there is adequate space in the HFS for this Portal data.

     The WebSphere Portal for z/OS Customization Dialog allocates a configuration HFS with a primary extent of 1600 cylinders by default during WebSphere Portal configuration. However, the default for the WebSphere

Application Server dialog is 250 cylinders. This allocation should be changed to have a primary extent of at least 1600 cylinders.

– As a file system, it is possible to use both HFS files and zFS aggregates, so be sure to allow both to expand.

  For zFS, this is allowed by the parameter `aggrgrow=on`. The command **`ioezadm configquery -all`** allows you to find out whether the option was set.

Example 2-1 zFS procedure

```
//ZFS      PROC REGSIZE=0M
//*
//ZFZGO    EXEC PGM=BPXVCLNY,REGION=&REGSIZE,TIME=1440
//*
//*STEPLIB DD DISP=SHR,DSN=hlq.SIOELMOD               <--ZFS LOADLIB
//IOEZPRM  DD DISP=SHR,                               <--ZFS PARM FILE
//  DSN=WTSCPLX1.IOE.&ZFSPARM..IOEFSZFS(IOEFSPRM)
      //*
```

  The option can be set dynamically (only valid for this IPL) by issuing **`ioezadm config -aggrgrow on`**. The option can be set permanently by allowing the zFS address space procedure to point to the parmlib member with this parameter. This is shown in Example 2-1 and Example 2-2.

Example 2-2 IOEFSPRM contents

```
aggrgrow=on
```

2. Update HTTP transport timeouts on the Deployment Manager.

   Before configuring WebSphere Portal for z/OS, you need to increase the HTTP transport timeout values on the Deployment Manager to prevent timeouts from occurring while deploying applications as follows:

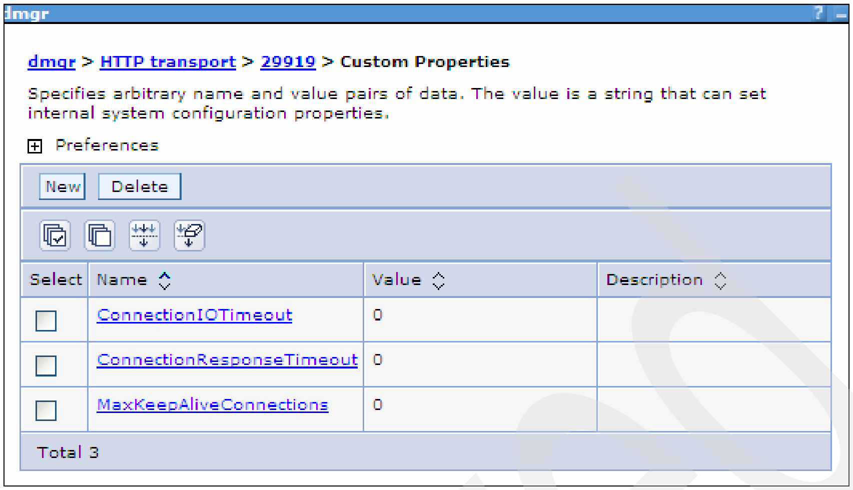
   – Start the administrative console and navigate to **System Administration** -> **Deployment Manager** -> **HTTP transport** -> the * for your port number -> **Custom Properties**; see Figure 2-3 on page 41.

   – For each HTTP transport listed for WebSphere Application Server, define the following variables:

     - ConnectionIOTimeout - the value should be 0.
     - ConnectionResponseTimeout - the value should be 0.
     - MaxKeepAliveConnections - the value should be 0.

     This change is made to both the secure and non-secure HTTP ports.

   – Recycle your Deployment Manager to pick up the changes.


dmgr > HTTP transport > 29919 > Custom Properties

Specifies arbitrary name and value pairs of data. The value is a string that can set internal system configuration properties.

⊞ Preferences

Select	Name	Value	Description
□	ConnectionIOTimeout	0	
□	ConnectionResponseTimeout	0	
□	MaxKeepAliveConnections	0	

Total 3

Figure 2-3 HTTP transport timeout values

3. Make sure the JVM max heap size for the Deployment Manager servant region is at least 512 MB.

   Go to the WebSphere Application Server administrative console and select **System Administration -> Deployment Manager -> dmgr > Process Definition -> Servant -> Java Virtual Machine.**

   The JVM maximum heap size should be set to 512 MB. If it is not, change it and then restart the DMGR before continuing with the Portal configuration.

2.3 Running the WebSphere Portal install dialog

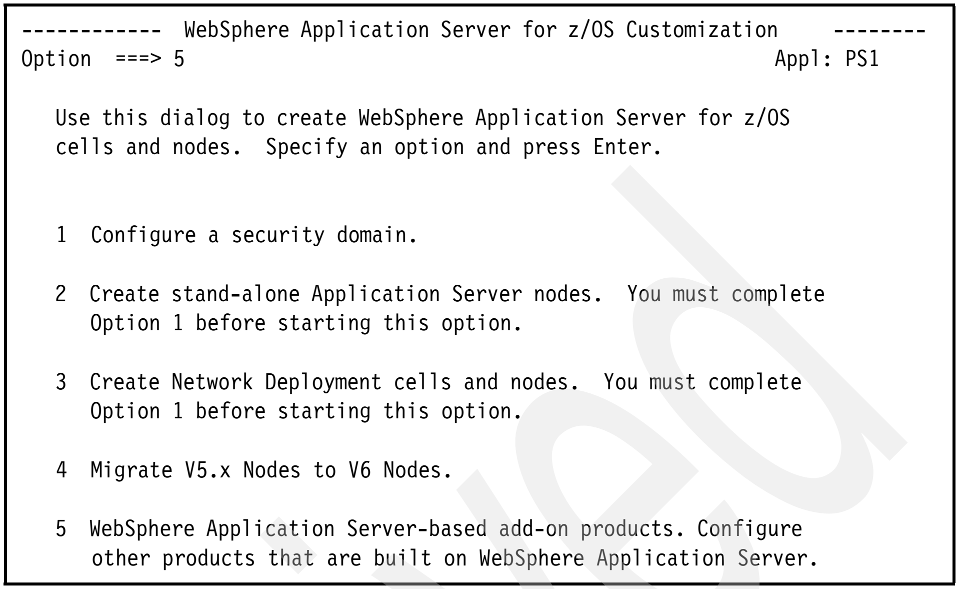
After WebSphere Application Server is fully configured with an empty node, enter the custom parameters for WebSphere Portal.

Go to ISPF option **6** and enter the command shown in Example 2-3 (replacing BBWP6049 with your high level qualifier) to start the install dialog.

Example 2-3 Starting the WebSphere customization dialog

```
exec 'BBWP6049.SBBOCLIB(BBOWSTRT)' 'APPL(PS1) PROD(EJP) PRODHLQ(BBWP6049)'
```

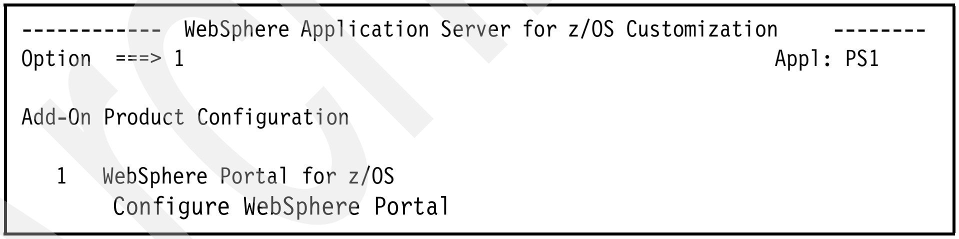
Choose option **5** to configure products that are built on to WebSphere Application Server, as shown in Figure 2-4 on page 42.



```
------------  WebSphere Application Server for z/OS Customization     --------
Option  ===> 5                                                 Appl: PS1

   Use this dialog to create WebSphere Application Server for z/OS
   cells and nodes.  Specify an option and press Enter.


   1  Configure a security domain.

   2  Create stand-alone Application Server nodes.  You must complete
      Option 1 before starting this option.

   3  Create Network Deployment cells and nodes.  You must complete
      Option 1 before starting this option.

   4  Migrate V5.x Nodes to V6 Nodes.

   5  WebSphere Application Server-based add-on products. Configure
      other products that are built on WebSphere Application Server.
```

Figure 2-4 Primary navigation panel

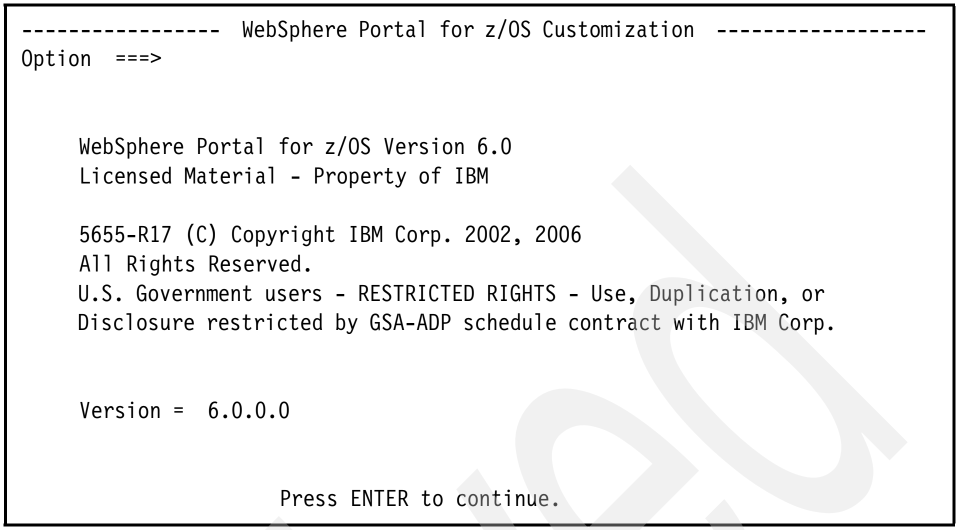
Choose `WebSphere Portal for z/OS` from the list of WebSphere add-on products, as shown in Figure 2-5. (We only had one choice in our scenario, but you may have more.)



```
------------  WebSphere Application Server for z/OS Customization     --------
Option  ===> 1                                                 Appl: PS1

Add-On Product Configuration

   1   WebSphere Portal for z/OS
        Configure WebSphere Portal
```

Figure 2-5 Choosing your WebSphere add-on product

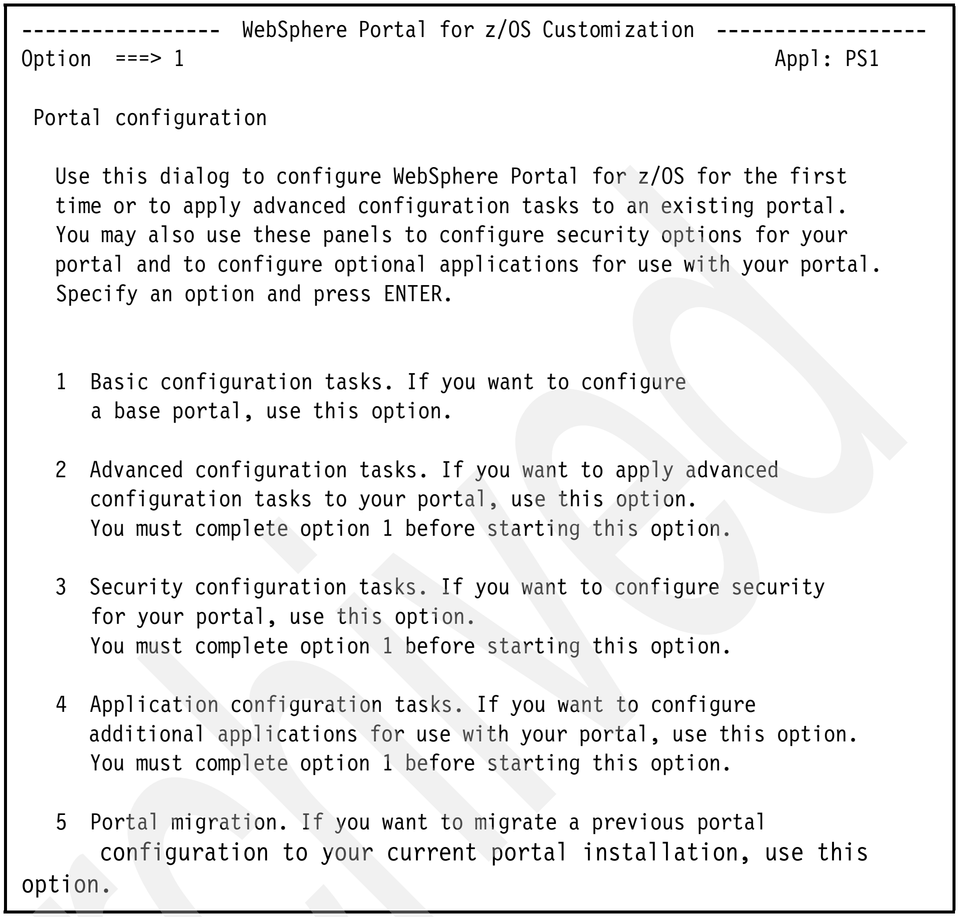
This panel will display each time you enter the install program. Verify the WebSphere Portal version before proceeding. Press Enter to accept the license agreement shown in Figure 2-6 on page 43.



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



     WebSphere Portal for z/OS Version 6.0
     Licensed Material - Property of IBM

     5655-R17 (C) Copyright IBM Corp. 2002, 2006
     All Rights Reserved.
     U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
     Disclosure restricted by GSA-ADP schedule contract with IBM Corp.


     Version =  6.0.0.0


                        Press ENTER to continue.
```

Figure 2-6 WebSphere Portal customization dialog - license panel for WebSphere Portal Version 6.0

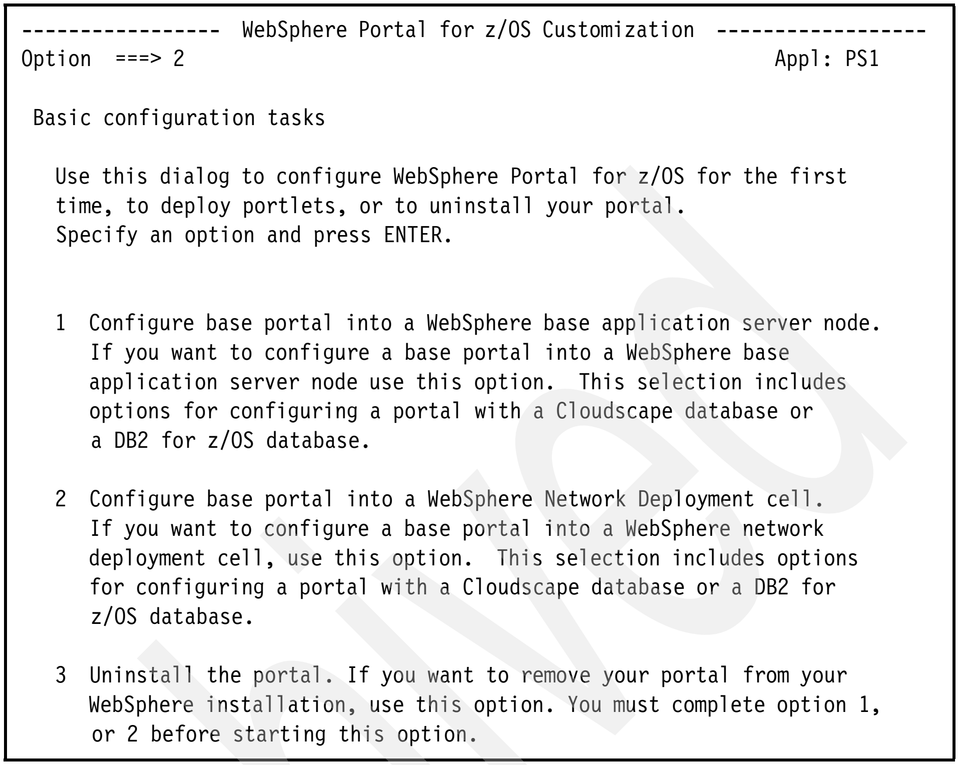
The panel shown in Figure 2-7 on page 44 will be used for all WebSphere Portal configuration tasks. Choose option **1** for basic installation of the product in the primary node.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                   Appl: PS1

 Portal configuration

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time or to apply advanced configuration tasks to an existing portal.
   You may also use these panels to configure security options for your
   portal and to configure optional applications for use with your portal.
   Specify an option and press ENTER.


   1  Basic configuration tasks. If you want to configure
      a base portal, use this option.

   2  Advanced configuration tasks. If you want to apply advanced
      configuration tasks to your portal, use this option.
      You must complete option 1 before starting this option.

   3  Security configuration tasks. If you want to configure security
      for your portal, use this option.
      You must complete option 1 before starting this option.

   4  Application configuration tasks. If you want to configure
      additional applications for use with your portal, use this option.
      You must complete option 1 before starting this option.

   5  Portal migration. If you want to migrate a previous portal
      configuration to your current portal installation, use this
option.
```

Figure 2-7 WebSphere Portal customization dialog - main WebSphere Portal customization screen

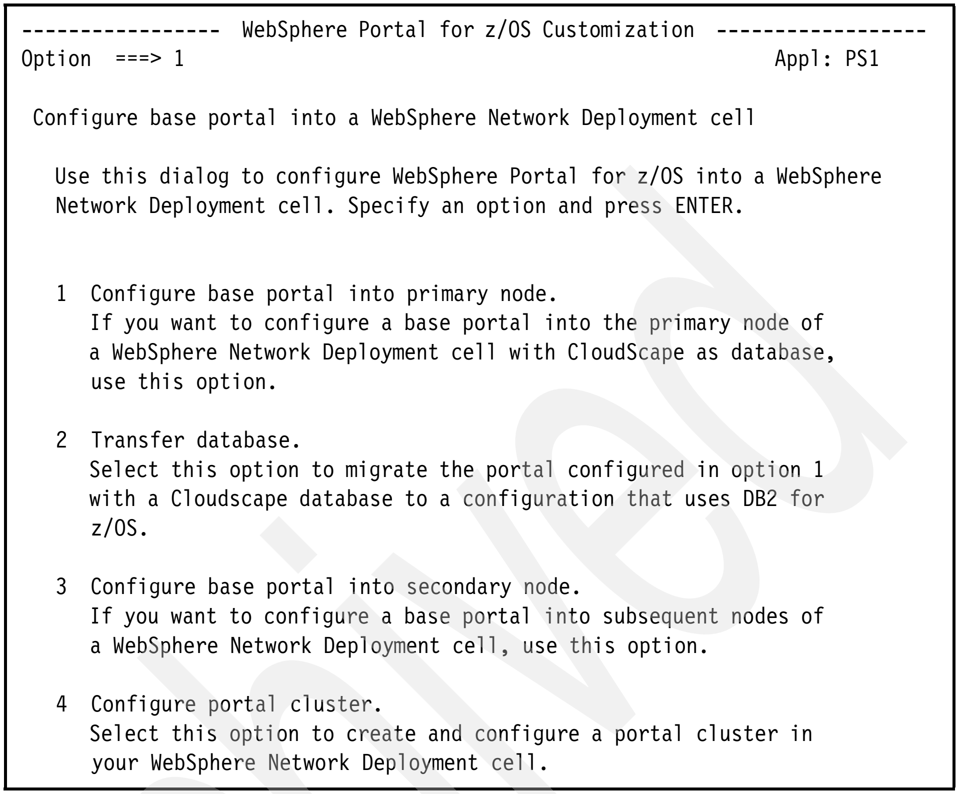
In the panel shown in Figure 2-8 on page 45 we selected option **2**, because in our scenario we are installing WebSphere Portal into a Network Deployment cell.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                    Appl: PS1

 Basic configuration tasks

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time, to deploy portlets, or to uninstall your portal.
   Specify an option and press ENTER.


   1  Configure base portal into a WebSphere base application server node.
      If you want to configure a base portal into a WebSphere base
      application server node use this option.  This selection includes
      options for configuring a portal with a Cloudscape database or
      a DB2 for z/OS database.

   2  Configure base portal into a WebSphere Network Deployment cell.
      If you want to configure a base portal into a WebSphere network
      deployment cell, use this option.  This selection includes options
      for configuring a portal with a Cloudscape database or a DB2 for
      z/OS database.

   3  Uninstall the portal. If you want to remove your portal from your
      WebSphere installation, use this option. You must complete option 1,
      or 2 before starting this option.
```

Figure 2-8 WebSphere Portal customization dialog - base WebSphere Application Server for Network Deployment

In the panel shown in Figure 2-9 on page 46, use option **1** to install WebSphere Portal into your primary WebSphere Application Server node. (Remember that we created an empty managed node earlier.)



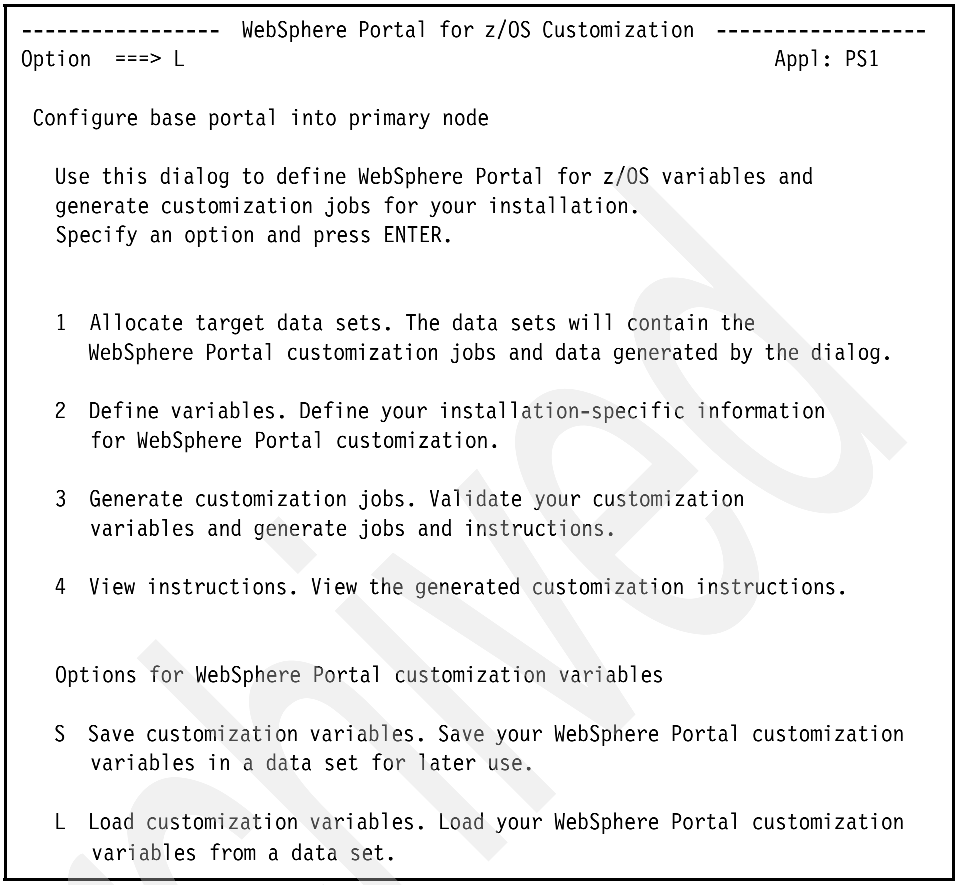
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                    Appl: PS1

 Configure base portal into a WebSphere Network Deployment cell

   Use this dialog to configure WebSphere Portal for z/OS into a WebSphere
   Network Deployment cell. Specify an option and press ENTER.


   1  Configure base portal into primary node.
      If you want to configure a base portal into the primary node of
      a WebSphere Network Deployment cell with CloudScape as database,
      use this option.

   2  Transfer database.
      Select this option to migrate the portal configured in option 1
      with a Cloudscape database to a configuration that uses DB2 for
      z/OS.

   3  Configure base portal into secondary node.
      If you want to configure a base portal into subsequent nodes of
      a WebSphere Network Deployment cell, use this option.

   4  Configure portal cluster.
      Select this option to create and configure a portal cluster in
      your WebSphere Network Deployment cell.
```

Figure 2-9 WebSphere Portal customization dialog - WebSphere Portal customization tasks

Values that have been previously used for the WebSphere Application Server installation can be loaded into the WebSphere Portal panels for convenience.

Make sure you load the variables from the SAVECFG data set where you saved the variables when configuring the empty node. This is important because the Portal will make use of many of the variables that apply to the node in which it is being configured. If you load incorrect variables or manually enter them again in subsequent panels, there is a greater possibility of a configuration error later.

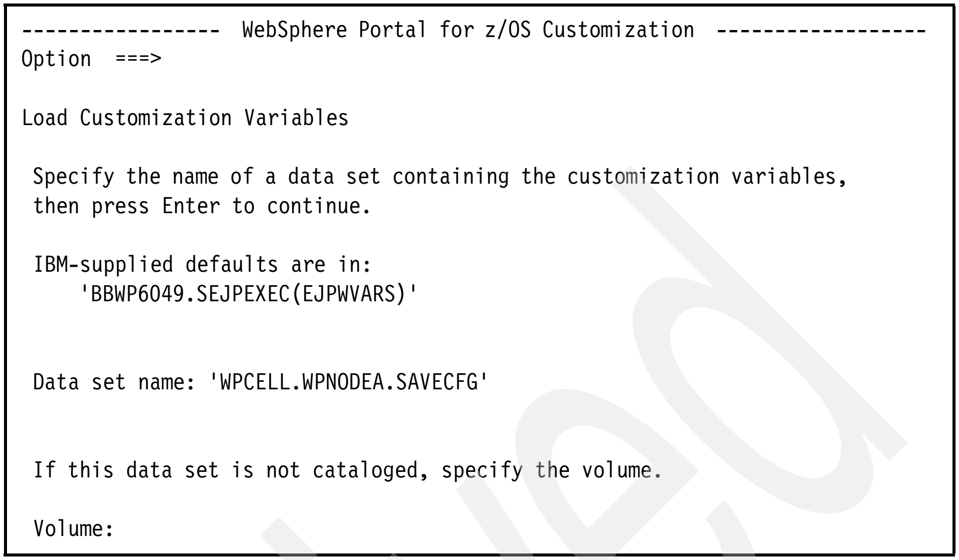
Choosing the option to load the variables is shown in Figure 2-10 on page 47.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> L                                                    Appl: PS1

 Configure base portal into primary node

   Use this dialog to define WebSphere Portal for z/OS variables and
   generate customization jobs for your installation.
   Specify an option and press ENTER.


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

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

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 2-10 WebSphere Portal customization dialog - load WebSphere Application Server variables

In the panel shown in Figure 2-11 on page 48, enter the data set where you saved your previous WebSphere Application Server variables. If no variables have been saved, the supplied default values will show up in the panels.



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

Load Customization Variables

 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in:
     'BBWP6049.SEJPEXEC(EJPWVARS)'


 Data set name: 'WPCELL.WPNODEA.SAVECFG'


 If this data set is not cataloged, specify the volume.

 Volume:
```

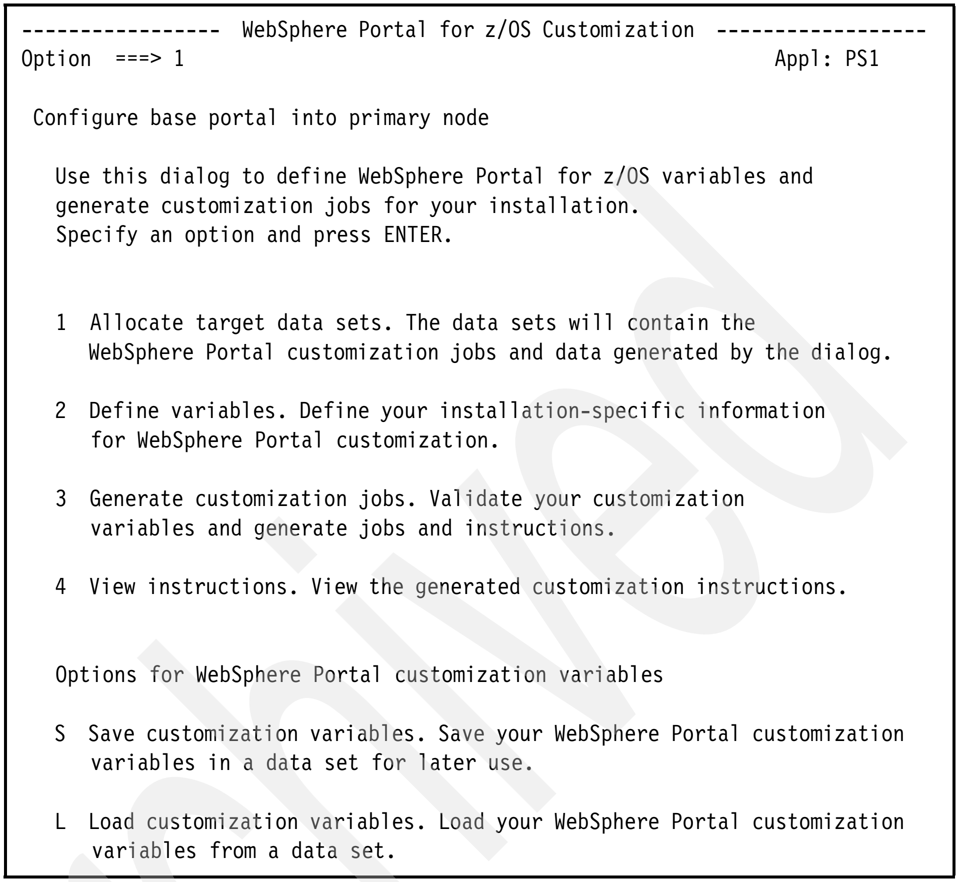
Figure 2-11 WebSphere Portal customization dialog - load WebSphere Application Server variables - data sets with variables

For more information about the use of variables and application profiles, refer to "Tips for using the customization dialog" in the WebSphere Portal for z/OS Infocenter at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/cu_runcust_zos.html

The WebSphere Portal dialog generates all the jobs that are required to customize the product. Data sets have to be allocated to hold all of these install jobs and the associated data.

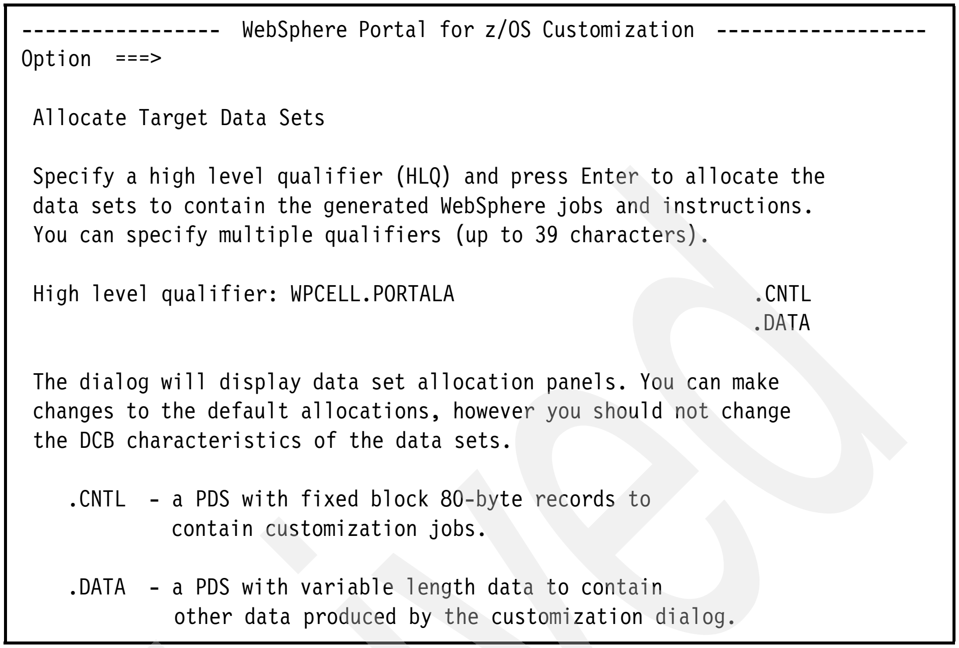
Choose option **1** in the panel shown in Figure 2-12 on page 49 to specify the name of these data sets.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                      Appl: PS1

 Configure base portal into primary node

   Use this dialog to define WebSphere Portal for z/OS variables and
   generate customization jobs for your installation.
   Specify an option and press ENTER.


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

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

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 2-12 WebSphere Portal customization dialog - allocate data sets

In the panel shown in Figure 2-13 on page 50, fill in the high level qualifier field with a qualifier that can be used for the install jobs. Accept the default characteristics for the data sets in the panels that follow.

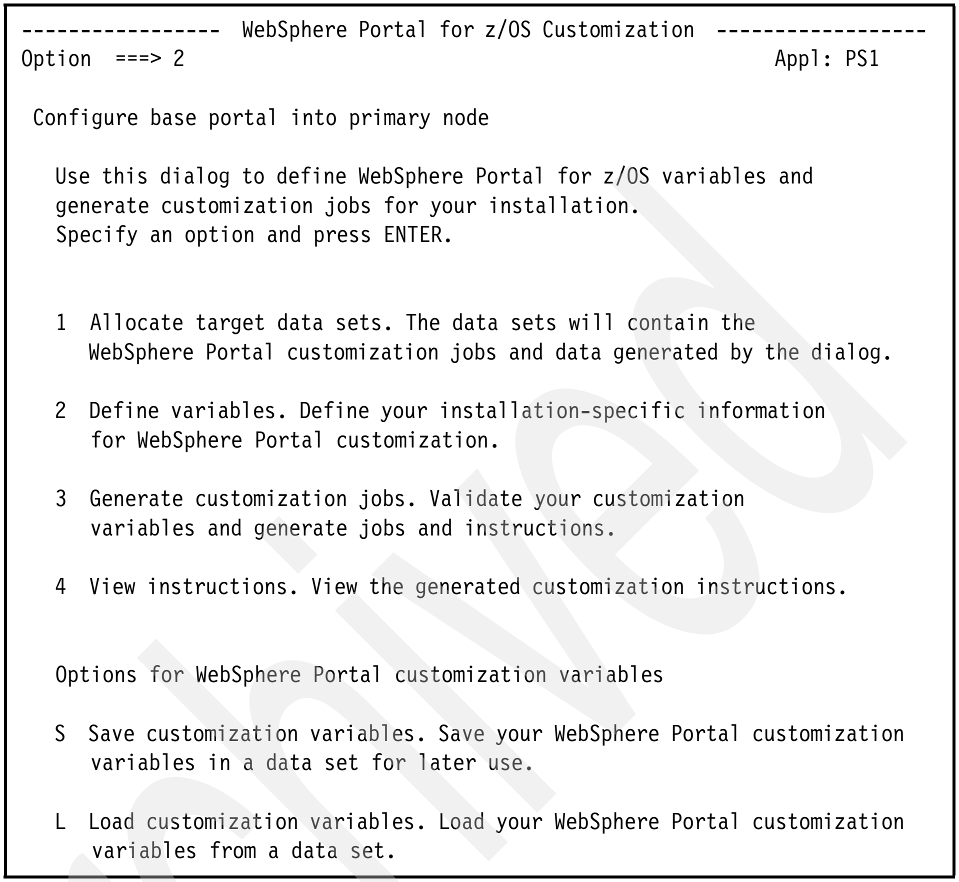


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 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: WPCELL.PORTALA                         .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 customization jobs.

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

Figure 2-13 WebSphere Portal customization dialog - HLQ of data sets

After the preparation for saving the configuration data is complete and the initial values are loaded, you start specifying the system values that WebSphere Portal requires.

In the panel shown in Figure 2-14 on page 51, choose option **2**: **Define variables**.

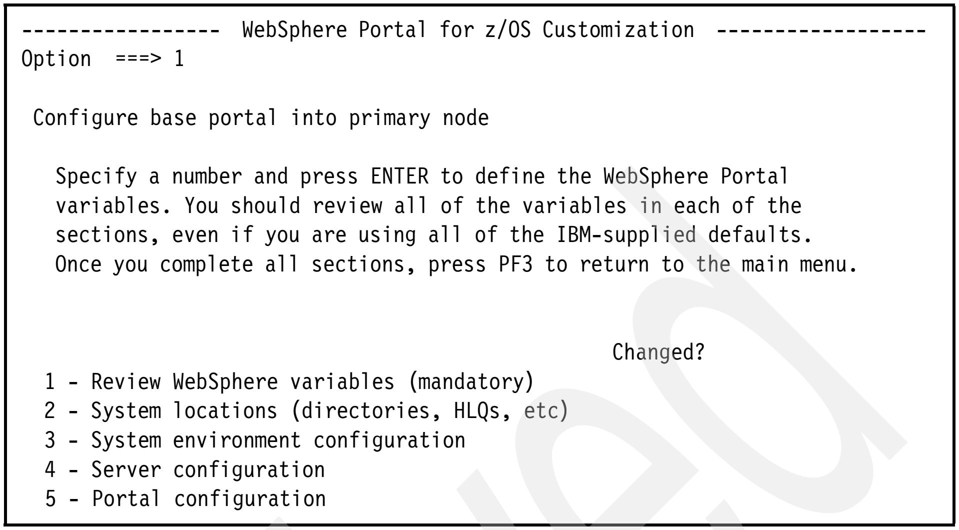


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                 Appl: PS1

 Configure base portal into primary node

   Use this dialog to define WebSphere Portal for z/OS variables and
   generate customization jobs for your installation.
   Specify an option and press ENTER.


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

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

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 2-14   WebSphere Portal customization dialog - define variables

In the next sequence of screens, we review all the variables entered during customization. Select option **1** in the panel shown in Figure 2-15 on page 52 to review the variables that were used for WebSphere Application Server.

If the values were not loaded from a file, then you will have to reenter them here.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)
  2 - System locations (directories, HLQs, etc)
  3 - System environment configuration
  4 - Server configuration
  5 - Portal configuration
```

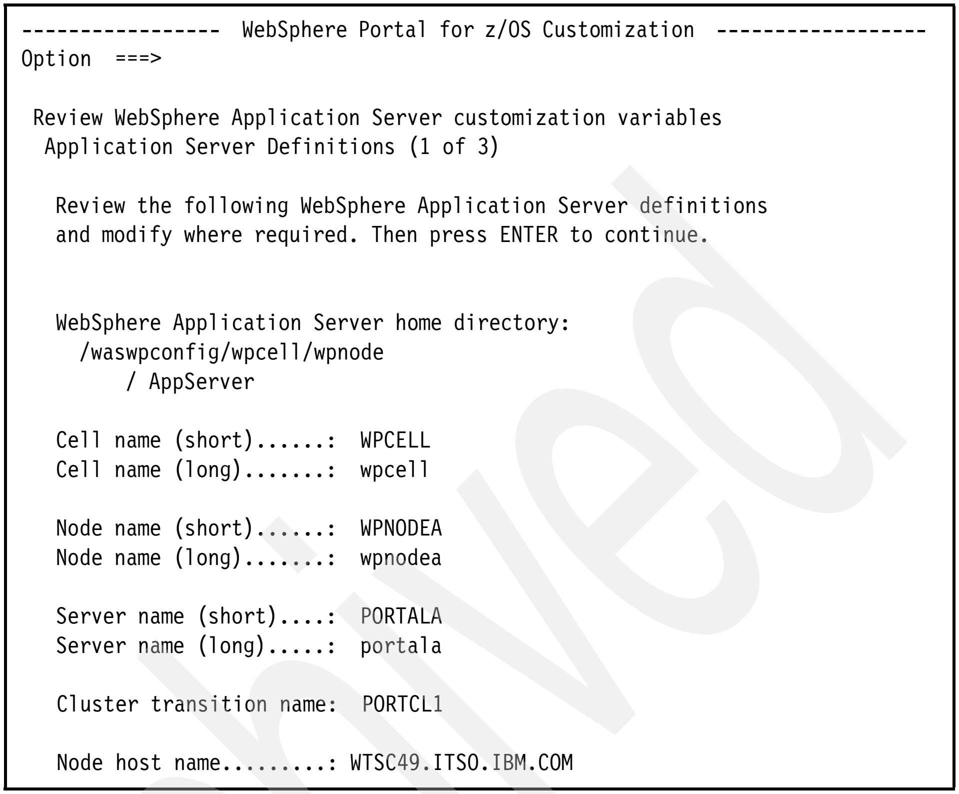
Figure 2-15 WebSphere Portal customization dialog - customize WebSphere Portal variables

The panel shown in Figure 2-16 on page 53 asks for existing WebSphere Application Server information. There are no new WebSphere Portal parameters here. The first parameter that is needed is the WebSphere Application Server “home” directory. This is where the base application server is installed; it is the “mount point” in the HFS system that contains the configuration.

In a Network Deployment install, it is the directory where you created your application server node. This field will be filled in already if previous ISPF variables were used. Set or verify the cell and node name from the WebSphere Application Server, as well.

The server name parameter is required even though we do not have any application servers defined in our empty node. This panel requires a server name (short), server name (long, and cluster transition name. This name is, in reality, the Application Environment (AE) name used by WLM for queuing work between control regions and servant regions. We use the same names here and in step 4.

Step 4 is where the WebSphere Portal Control region name will be specified. The node host name should be the domain name of the LPAR you are running on. Do not use the DVIPA name if a Sysplex Distributor is being used. Using the DVIPA name causes communication errors when a secondary node is added to the Deployment Manager.



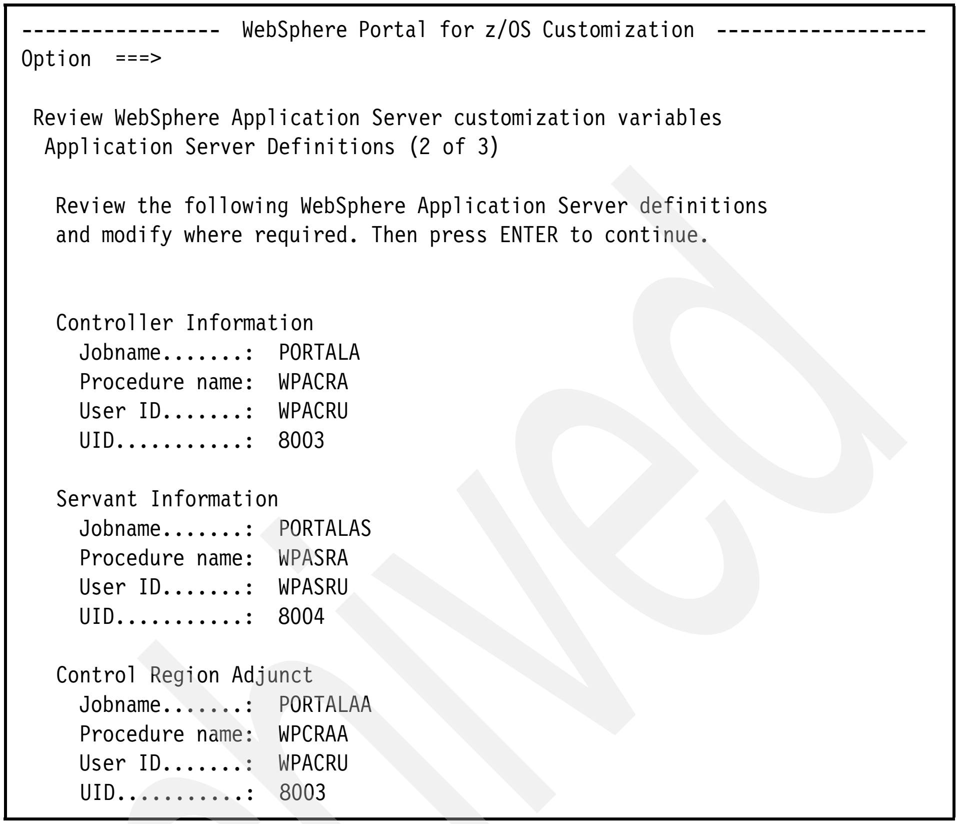
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===>

 Review WebSphere Application Server customization variables
  Application Server Definitions (1 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


   WebSphere Application Server home directory:
     /waswpconfig/wpcell/wpnode
         / AppServer

   Cell name (short)......:  WPCELL
   Cell name (long).......:  wpcell

   Node name (short)......:  WPNODEA
   Node name (long).......:  wpnodea

   Server name (short)....:  PORTALA
   Server name (long).....:  portala

   Cluster transition name:  PORTCL1

   Node host name.........: WTSC49.ITSO.IBM.COM
```

Figure 2-16 WebSphere Portal customization dialog - customize WebSphere Portal variables (1 of 3)

In the panel shown in Figure 2-17 on page 54, the jobname will be copied from the previous panel. The started task procedure name, started task RACF user ID and USS UID associated with the RACF ID must be specified in this panel.

The IDs in this panel must be defined in RACF *before* executing the batch jobs. They must also have an OMVS UID with an OMVS home directory; failure to provide this will cause many of the batch jobs to fail, because they write output to OMVS.

Other RACF permits and definitions have to be done, at the WebSphere Application Server level.

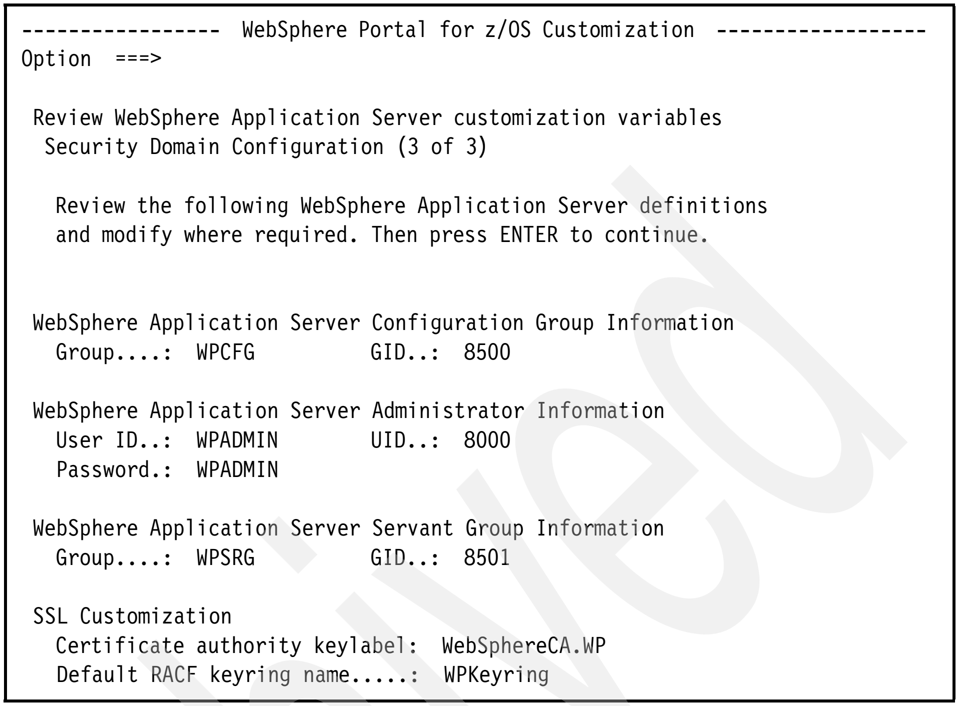


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Review WebSphere Application Server customization variables
  Application Server Definitions (2 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


   Controller Information
     Jobname.......:  PORTALA
     Procedure name:  WPACRA
     User ID.......:  WPACRU
     UID...........:  8003

   Servant Information
     Jobname.......:  PORTALAS
     Procedure name:  WPASRA
     User ID.......:  WPASRU
     UID...........:  8004

   Control Region Adjunct
     Jobname.......:  PORTALAA
     Procedure name:  WPCRAA
     User ID.......:  WPACRU
     UID...........:  8003
```

Figure 2-17 Customize WebSphere Portal variables (2 of 3)

If you loaded the variables that you had saved when configuring the empty node on the panel shown in Figure 2-10 on page 47, then all the user IDs, groups, UIDs and GIDs on the panel in Figure 2-18 on page 55 should already be correct for this cell.

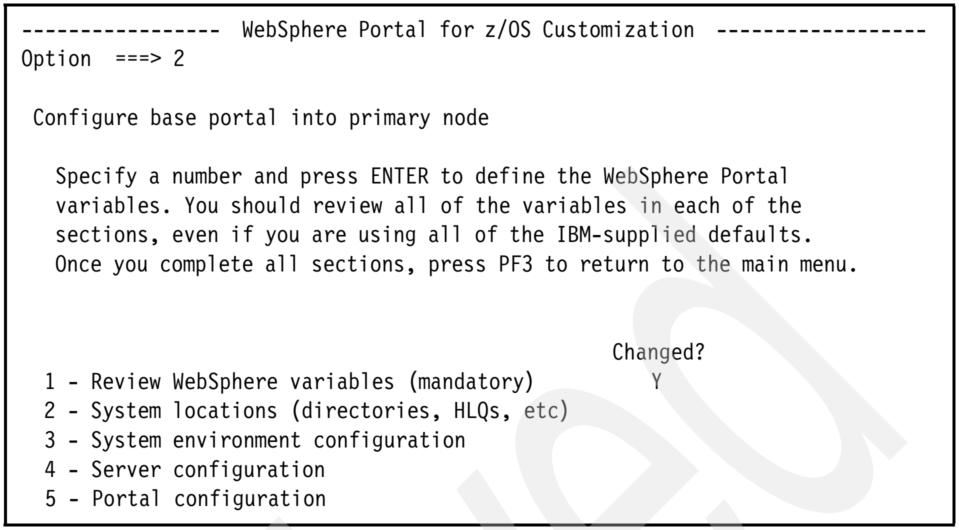
However, if you did not load those variables and you are typing in the values again, it is important that you enter the *same* values that were used when configuring the Deployment Manager and the empty node. (We strongly recommend that you load the correct saved variables, rather than type them in again here.)



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

 Review WebSphere Application Server customization variables
  Security Domain Configuration (3 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


 WebSphere Application Server Configuration Group Information
   Group....:  WPCFG         GID..:  8500

 WebSphere Application Server Administrator Information
   User ID..:  WPADMIN       UID..:  8000
   Password.:  WPADMIN

 WebSphere Application Server Servant Group Information
   Group....:  WPSRG         GID..:  8501

 SSL Customization
   Certificate authority keylabel:  WebSphereCA.WP
   Default RACF keyring name.....:  WPKeyring
```

Figure 2-18 WebSphere Portal customization dialog - customize WebSphere Portal variables (3 of 3)

The main configuration screen will now show that all WebSphere variables have been reviewed by displaying `Y` in the `Changed?` column. Select option **2** in the panel shown in Figure 2-19 on page 56.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)           Y
  2 - System locations (directories, HLQs, etc)
  3 - System environment configuration
  4 - Server configuration
  5 - Portal configuration
```

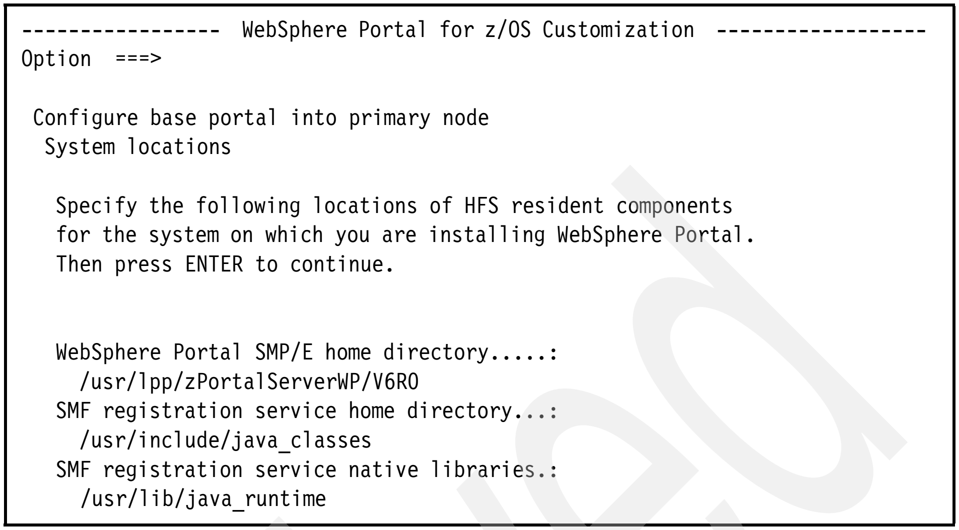
Figure 2-19 WebSphere Portal customization dialog - customize WebSphere Portal variables (reviewing system locations)

In the panel shown in Figure 2-20 on page 57 the SMP/E home directory is set to the directory where the WebSphere Portal files had been placed during the SMP/E process. The *SMF registration service* provides for usage-based charging of Java-based system software like WebSphere Portal. Without these classes WebSphere Portal will not initialize.

The SMF registration service is a software prerequisite that is part of z/OS 1.7 or later. With z/OS 1.6 you need to install the PTFs mentioned in the software prerequisites in 1.4.2, “Software prerequisites” on page 15.

The panel shown in Figure 2-20 on page 57 requests information about the USS directories used on the system. The first value is the USS directory where SMP/E copied the WebSphere Portal code.

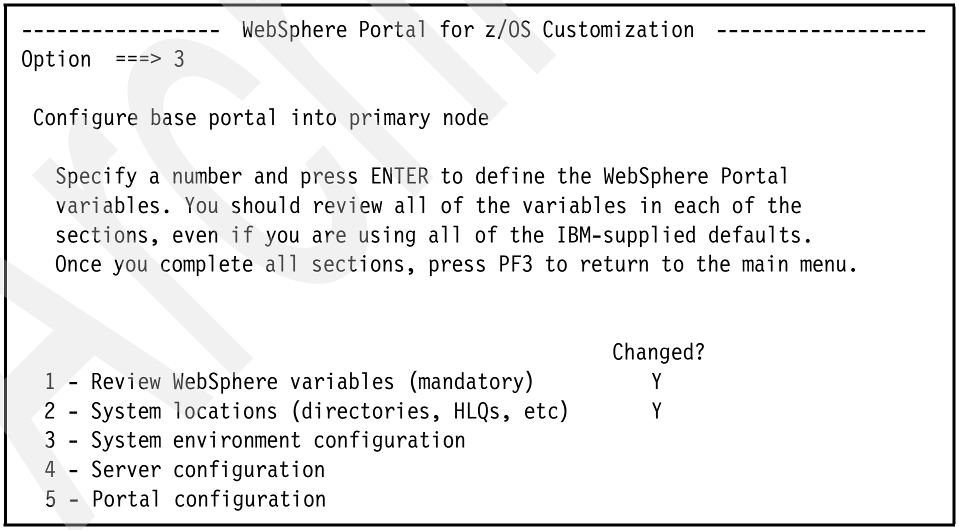
You can locate “SMF registration service home directory” by searching the HFS for files ifaedjreg.jar and ifaedregDoc.jar. You can locate “SMF registration service native libraries” by searching the HFS for libifaedjreg.so and libifaedjreg64.so.



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

 Configure base portal into primary node
  System locations

   Specify the following locations of HFS resident components
   for the system on which you are installing WebSphere Portal.
   Then press ENTER to continue.


   WebSphere Portal SMP/E home directory.....:
     /usr/lpp/zPortalServerWP/V6R0
   SMF registration service home directory...:
     /usr/include/java_classes
   SMF registration service native libraries.:
     /usr/lib/java_runtime
```

Figure 2-20 Customize WebSphere Portal variables (specifying locations)

The main configuration screen will now show that all WebSphere variables and System locations have been reviewed by displaying `Y` in the `Changed?` column. Select option **3** in the panel shown in Figure 2-21.



```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===> 3

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)          Y
  2 - System locations (directories, HLQs, etc)       Y
  3 - System environment configuration
  4 - Server configuration
  5 - Portal configuration
```

Figure 2-21 Customize WebSphere Portal variables (system environment configuration)

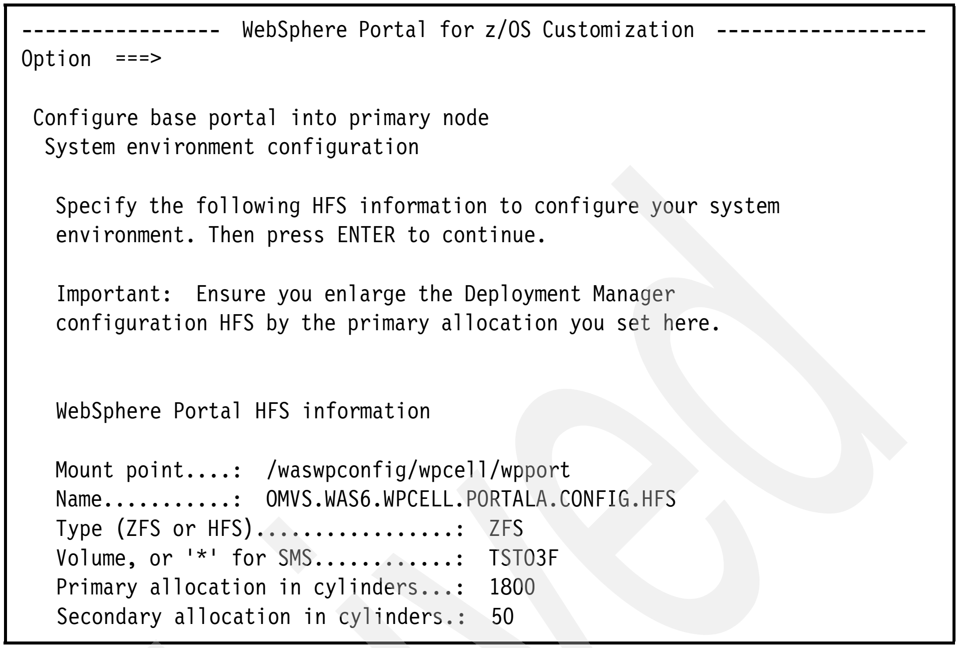
In the panel shown in Figure 2-22 on page 59, the mount point of WebSphere Portal is the directory where all of your WebSphere Portal customized code will be placed. The Name field refers to the data set that will be mounted at that directory point.

If a file system type of zFS is chosen, it must be set to allow dynamic growth by using the `aggrgrow=on` parameter, as shown in Example 2-2 on page 40. (You can find other temporary solutions to enabling dynamic growth in “IBM WebSphere Portal Enable for z/OS Version 6.0 Release Notes” if IOEFSPRM cannot be updated immediately.)

We recommend that you use at least 1600 cylinders for your primary allocation on the WebSphere Portal HFS/zFS data set. The size of the Portal configuration zFS determines how much extra space (above the default) has to be added to the Deployment Manager's zFS.

The Portal configuration zFS needs to be at least 1600 cylinders, and the DMGRs zFS needs to be increased by at least 1600 cylinders to hold the master copy of the Portal configuration. Thus, typically a DMGR might need a zFS of 1800 cylinders (200+1600), and the Portal configuration zFS must be 1600 cylinders for each node.

Also note that there is also a zFS for the WebSphere Application Server node itself. That can be allowed to start with the default value of 250 cylinders. Make sure the volumes that you enter have enough space. When not using SMS, you can use a comma (,) to separate multiple volumes.



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

 Configure base portal into primary node
  System environment configuration

   Specify the following HFS information to configure your system
   environment. Then press ENTER to continue.

   Important:  Ensure you enlarge the Deployment Manager
   configuration HFS by the primary allocation you set here.


   WebSphere Portal HFS information

   Mount point....:  /waswpconfig/wpcell/wpport
   Name...........:  OMVS.WAS6.WPCELL.PORTALA.CONFIG.HFS
   Type (ZFS or HFS).................:  ZFS
   Volume, or '*' for SMS............:  TST03F
   Primary allocation in cylinders...:  1800
   Secondary allocation in cylinders.:  50
```

Figure 2-22 WebSphere Portal customization dialog - customize WebSphere Portal variables (specify HFS information)

Currently we are building a Portal in the primary node. Later we will build clones on other nodes in other LPARs and join them in a cluster. HFS/zFS configuration files are also required for those secondary nodes and clones.

We would like to use the same USS paths on all LPARS. To achieve this goal, we will structure the file mounting via symbolic links, knowing that we have at our disposal the symbolic parameter &SYSNAME, which represents the name of the LPAR in the sysplex. Figure 2-23 shows how it is organized.

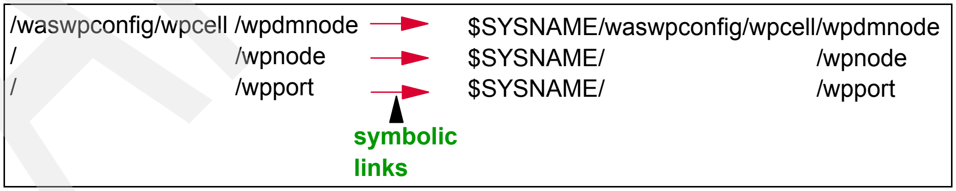


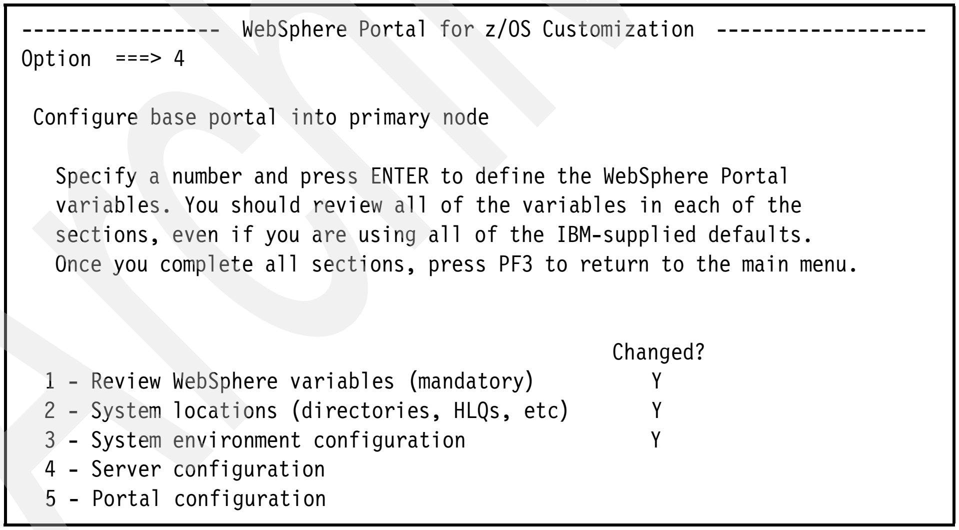
Figure 2-23 HFS organization with symbolic links

Tip: The zFS/HFS datasets for each node should *only* be mounted on the LPAR where that node resides. Why? Because in the event of an LPAR failure, System Automation software may automatically move the mount point of a zFS/HFS to another LPAR. After re-IPLing the failed LPAR, the Portal can take a very long time to initialize because its zFS/HFS is owned by the wrong LPAR.

To avoid this possibility, specify the NOAUTOMOVE and SYSNAME(aaaa) parameters (where aaaa is the LPAR name) on the MOUNT statements for WebSphere and Portal configuration zFS/HFS datasets in BPXPRMxx. This will ensure they are only mounted on the correct LPAR.

Do not specify SYSNAME(aaaa) and NOAUTOMOVE for the Deployment Manager's zFS/HFS, because you may want to start the DMGR on any LPAR in the sysplex.

The main configuration screen will now show that all WebSphere variables, System locations, and the System environment configuration have been reviewed by displaying `Y` in the `Changed?` column. Select option **4** in the panel shown in Figure 2-24.



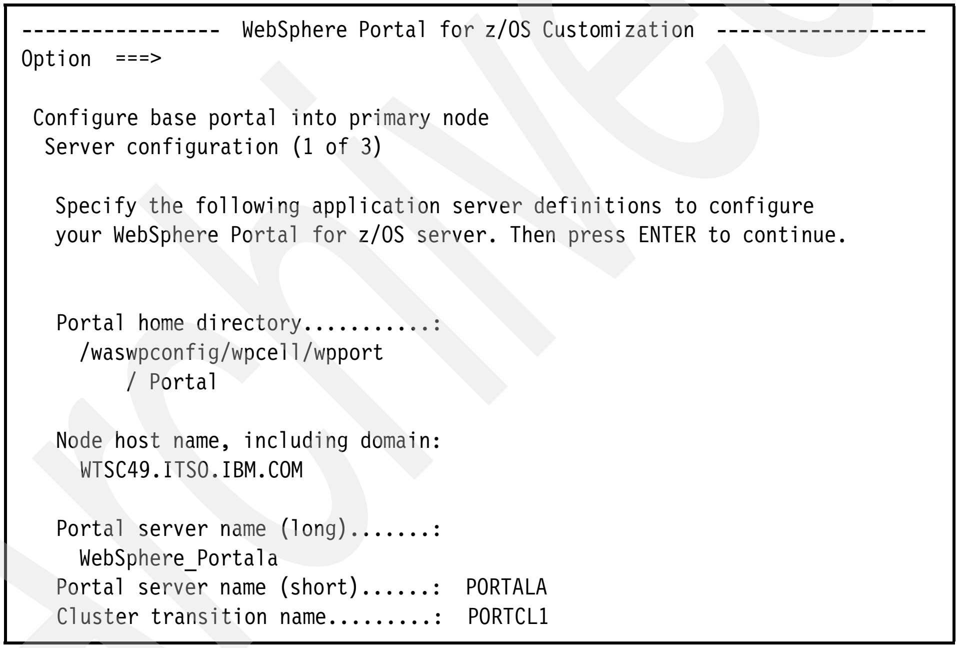
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 4

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)           Y
  2 - System locations (directories, HLQs, etc)        Y
  3 - System environment configuration                 Y
  4 - Server configuration
  5 - Portal configuration
```

Figure 2-24 WebSphere Portal customization dialog - customize WebSphere Portal variables (choose server configuration)

The mount point from the panel shown in Figure 2-24 on page 60 will be automatically plugged into the WebSphere Portal home directory field of the panel shown in Figure 2-25. Most installations add a directory to the path called *Portal* to clarify what is contained in that directory. The domain name of the WebSphere Portal is entered under `Node host name`.

The final three parameters are the names that are used for the new application server that will be defined for WebSphere Portal, as explained here:

- The *long name* is the name that appears under application servers in the WebSphere Application Server administrative console.
- The *short name* is used for the WebSphere Portal address space.
- The *cluster transition name* is the name of the process that is used for the WebSphere Portal Control region to communicate with its servant region. It is also called the *Application Environment*.



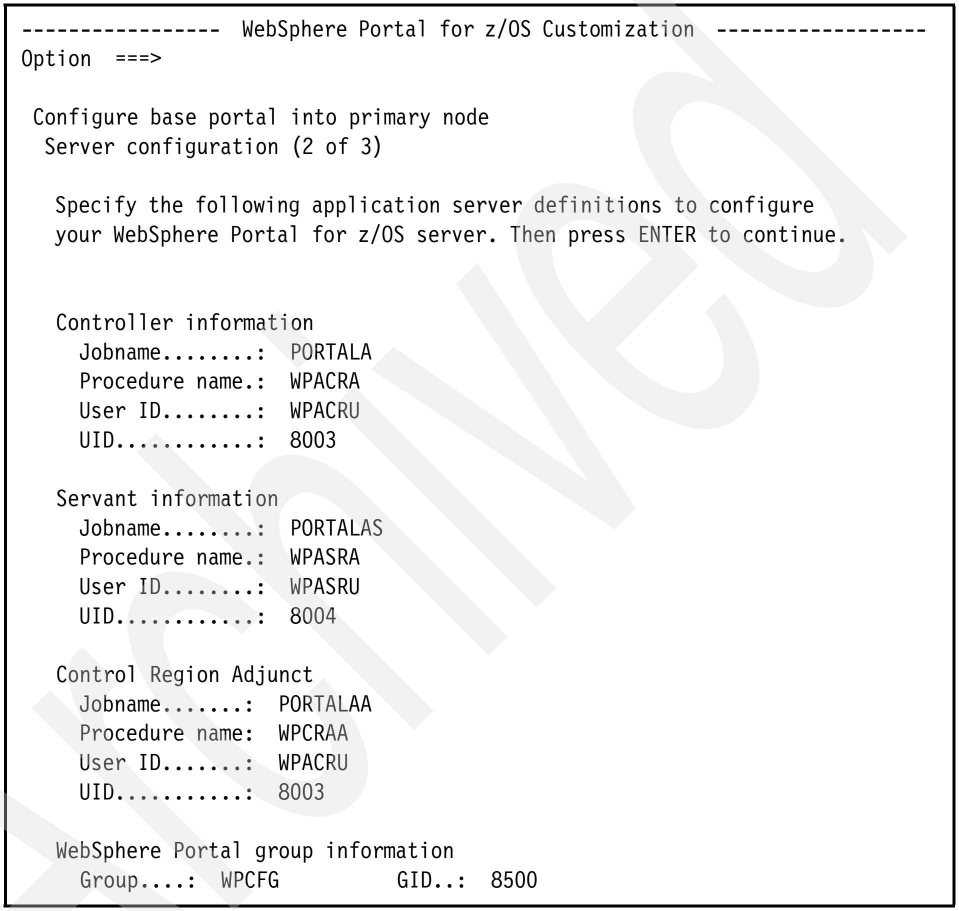
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Configure base portal into primary node
  Server configuration (1 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   Portal home directory...........:
     /waswpconfig/wpcell/wpport
         / Portal

   Node host name, including domain:
     WTSC49.ITSO.IBM.COM

   Portal server name (long).......:
     WebSphere_Portala
   Portal server name (short)......:  PORTALA
   Cluster transition name.........:  PORTCL1
```

Figure 2-25 WebSphere Portal customization dialog - customize WebSphere Portal variables (server configuration - 1of 3)

The node host name should point to a non-distributed TCP/IP address, and not to a DVIPA address.

The job names on the panel shown in Figure 2-26 were generated from the values in the panel shown in Figure 2-25 on page 61, as explained here:

- *Procedure name* is the name of the started task procedure’s member in SYS1.PROCLIB.
- *User ID* is the ID that is used to run the associated started task while UID is the USS UID given to the omvs segment of the MVS™ user ID.



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

 Configure base portal into primary node
  Server configuration (2 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   Controller information
     Jobname........:  PORTALA
     Procedure name.:  WPACRA
     User ID........:  WPACRU
     UID............:  8003

   Servant information
     Jobname........:  PORTALAS
     Procedure name.:  WPASRA
     User ID........:  WPASRU
     UID............:  8004

   Control Region Adjunct
     Jobname.......:  PORTALAA
     Procedure name:  WPCRAA
     User ID.......:  WPACRU
     UID...........:  8003

   WebSphere Portal group information
     Group....:  WPCFG          GID..:  8500
```

Figure 2-26 WebSphere Portal customization dialog - customize WebSphere Portal variables (server configuration 2 of 3)

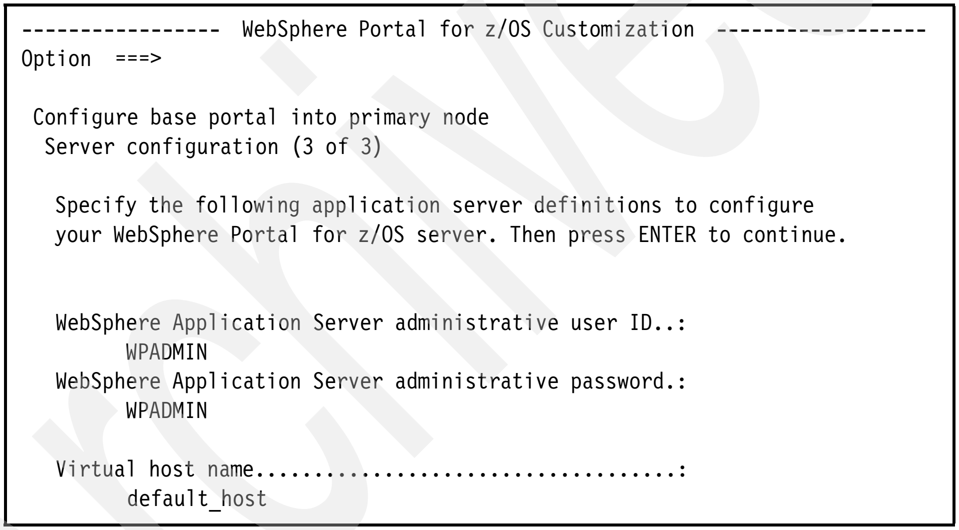
The number of PROCs used by WebSphere Application Server is rather limited. We have a separate one for the daemon control region, Deployment Manager control region and deployment servant region. The control region PROC for the application server is common, and also used by the Node Agents. The servant PROC is also common for all application server servants.

In order for WebSphere Portal to install correctly, many of the jobs must run with the WebSphere Application Server administrative ID, as shown in Figure 2-27.

The ID and password that are entered here will be appended to the end of the job card for jobs that require administrative authority in WebSphere Application Server.

As displayed in the panel:

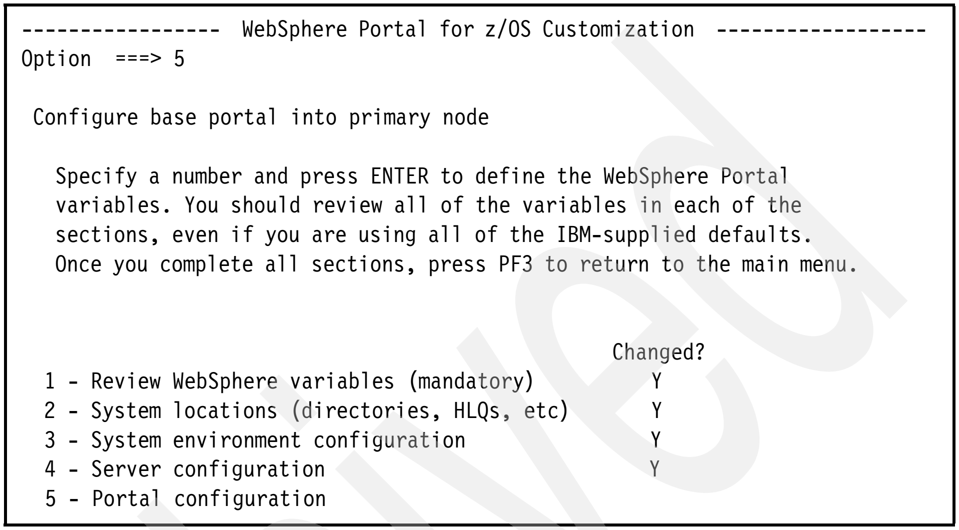
- `Virtual host name` is the name of the assembly of TCP/IP DNS name and port number combinations from which we can access any applications in the cell including the WebSphere Administrative Console application.
- The `default_host` virtual host, which is always present, will be updated with additional hostname/port combinations by the EJPSDNC1 job that configures the Portal server in the node.



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

 Configure base portal into primary node
  Server configuration (3 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   WebSphere Application Server administrative user ID..:
         WPADMIN
   WebSphere Application Server administrative password.:
         WPADMIN

   Virtual host name....................................:
         default_host
```

Figure 2-27 WebSphere Portal customization dialog - customize WebSphere Portal variables (server configuration 3 of 3)

The main configuration screen will now show that all WebSphere variables, system locations, system environment configuration and server configuration have been reviewed by displaying `Y` in the `Changed?` column. Select option **5** in the panel shown in Figure 2-28.



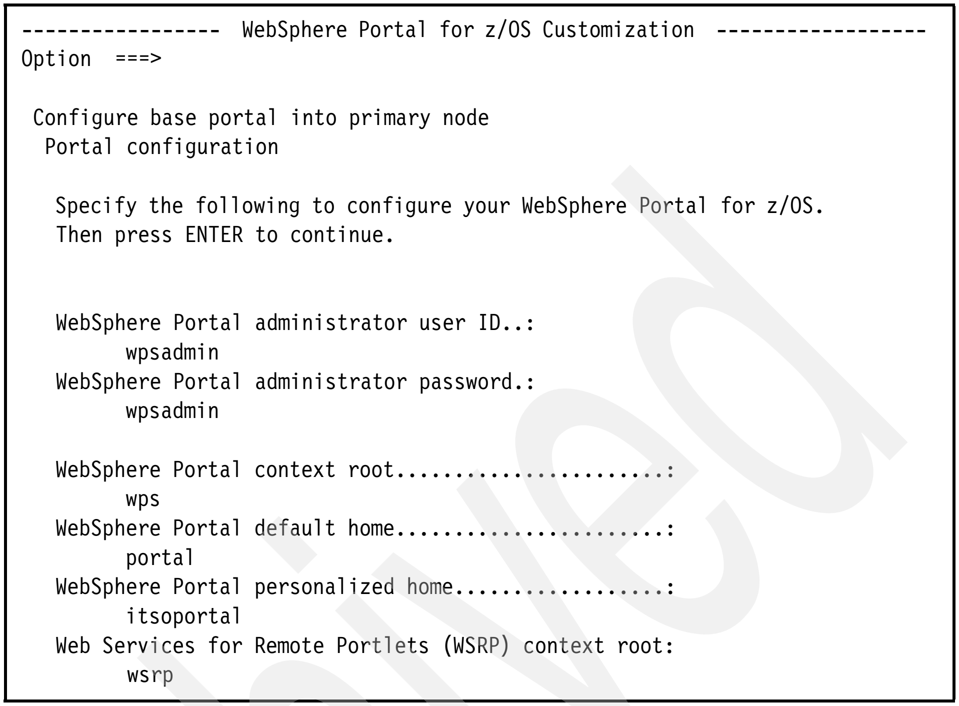
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===> 5

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)          Y
  2 - System locations (directories, HLQs, etc)       Y
  3 - System environment configuration                Y
  4 - Server configuration                            Y
  5 - Portal configuration
```

Figure 2-28 Customize WebSphere Portal variables (portal configuration)

The next panel, shown in Figure 2-29 on page 65, defines several settings for the WebSphere Portal application server. The WebSphere Portal administrator ID and password can be set to anything. In later chapters, when we prepare the security enablement, we discuss using an ID located in an external LDAP registry as the Portal administrator.

As displayed in the panel:

- `WebSphere Portal context root` is a value that will be reserved for the beginning of all URLs that access the portal; for example: `http://localhost:29818/wps/portal`
- `Default home` is the URL that the WebSphere Portal points to before a user is authenticated; for example: `http://localhost:29818/wps/portal`
- `Personalized home` is the URL that users are directed to after they authenticate with the portal.
- `WSRP context root` is the URL used to access remote portlets.



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

 Configure base portal into primary node
  Portal configuration

   Specify the following to configure your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   WebSphere Portal administrator user ID..:
         wpsadmin
   WebSphere Portal administrator password.:
         wpsadmin

   WebSphere Portal context root.......................:
         wps
   WebSphere Portal default home.......................:
         portal
   WebSphere Portal personalized home..................:
         itsoportal
   Web Services for Remote Portlets (WSRP) context root:
         wsrp
```

Figure 2-29 WebSphere Portal customization dialog - customize WebSphere Portal variables (portal configuration - settings)

The default for `WebSphere Portal personalized home` is myportal. If you change this default to another value (as we did), an additional job (EJPSPTHI) must be run.

However, this job was not part of the generated items at the time of writing; refer to the following note for more information about this topic.

Changing context root, default home, or personalized home in the panel shown in Figure 2-29:

If these values are changed to anything other than the supplied defaults, job EJPSPTHI will have to be generated through the install panels to activate the customized URIs. Generate and run this job after all other WebSphere Portal install jobs have been completed.

This job changes the WebSphere Portal URI for any of the custom names specified in these three fields (in our case, `itsoportal`). Simply changing the WpsPersonalizedHome variable in wpconfig.properties will *not* change the URI.

This custom job updates each WebSphere Portal application with the new URI, and recycles the application server. Be aware that it takes a considerable amount of time to run because it updates *every* WebSphere Portal application.

If this job is not run when creating a custom personalized home URI, then the following error message will be displayed in the browser window when you log on to the portal:

```
SRVE019E: File not found: /itsoportal/!ut/p/c1/0wcA1NLTeQ!!
```

This error only occurs in the Mozilla Firefox browser. Internet Explorer shows the error `page not found`, with no other information.

A URL decoding error will also be displayed in the WebSphere Portal Servant address space:

```
URL could not be de decoded.java.io.IOException
```

To generate EJPSPTHI, select option **2** from the WebSphere Portal install dialog:

```
2  Advanced configuration tasks. If you want to apply advanced
   configuration tasks to your portal, use this option.
   You must complete option 1 before starting this option.
```

Then, choose option **2** to configure portal paths:

```
2  Configure portal paths. Select this option to modify the directories
   that are used to serve public and private data from your portal.
```

Next, select option **2** to define variables:

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

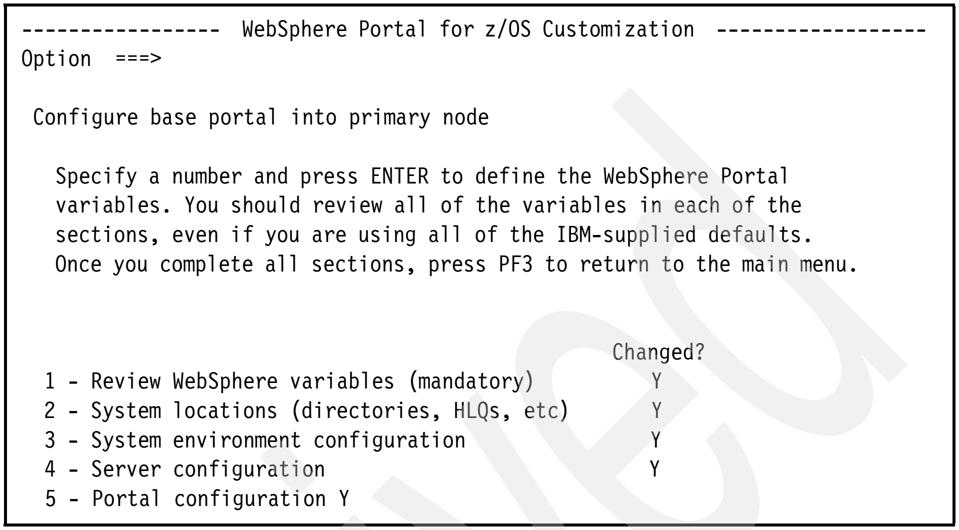
Ensure the value that was changed in the panel shown in Figure 2-29 on page 65 is also changed here.

Verify all your variables are correct on the next panel and then generate the job for this task:

```
3  Generate customization jobs. Validate your choices
   and generate jobs and instructions.
```

Finally, stop your WebSphere_Portal application server and run EJPSPTHI. The job should return RC=0.

The main configuration screen will now show that all categories have been updated with custom settings.

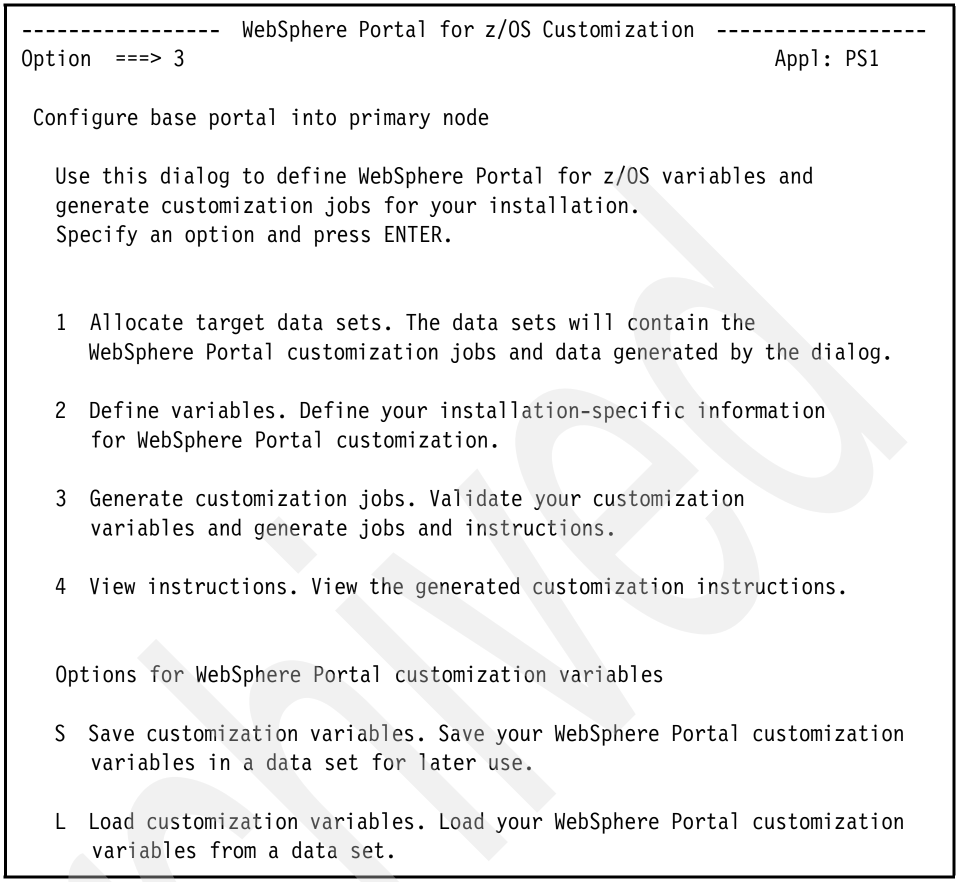


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Configure base portal into primary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)         Y
  2 - System locations (directories, HLQs, etc)      Y
  3 - System environment configuration               Y
  4 - Server configuration                           Y
  5 - Portal configuration Y
```

Figure 2-30 WebSphere Portal customization dialog - customize WebSphere Portal variables (review variables)

Press F3 to return to the main menu. At the main menu, save your customization variables by selecting the **S** option.

After all the system settings are entered, the installation jobs can be generated by selecting option **3** in the panel shown in Figure 2-31 on page 68.

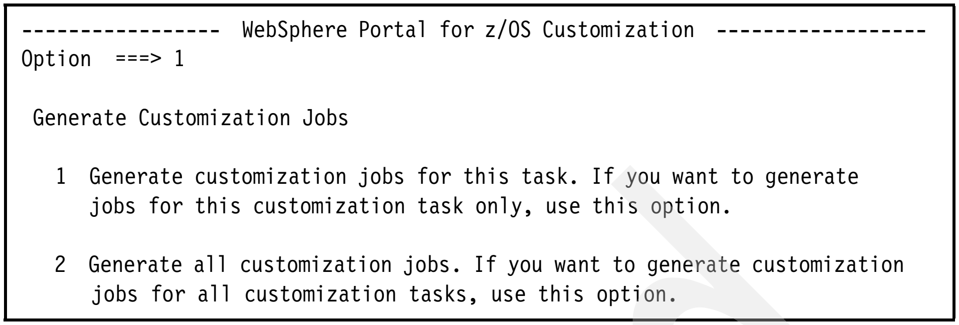


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 3                                                     Appl: PS1

 Configure base portal into primary node

   Use this dialog to define WebSphere Portal for z/OS variables and
   generate customization jobs for your installation.
   Specify an option and press ENTER.


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

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

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 2-31 WebSphere Portal customization dialog - generate installation jobs

The installation jobs for *all* WebSphere Portal customization tasks can be generated here—or you can choose to generate only the jobs that relate to the current customization task. (In our case, we only generated the jobs for the current customization task.)

Select option **1** in the panel shown in Figure 2-32 on page 69.



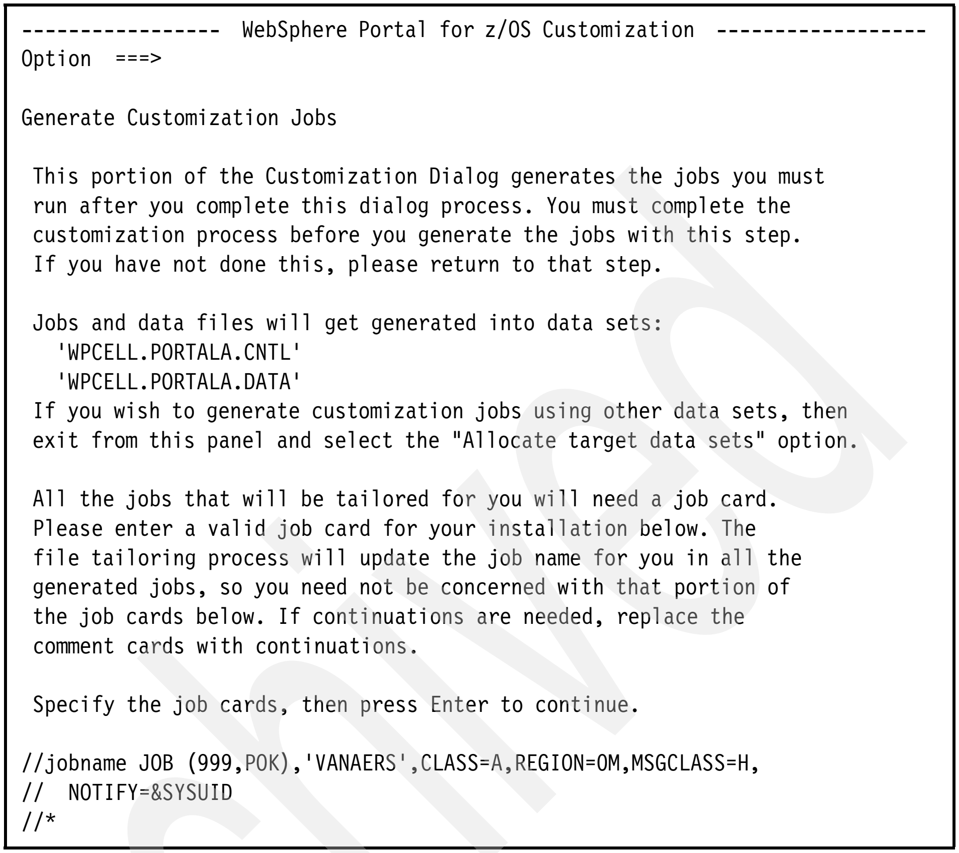
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===> 1

 Generate Customization Jobs

   1  Generate customization jobs for this task. If you want to generate
      jobs for this customization task only, use this option.

   2  Generate all customization jobs. If you want to generate customization
      jobs for all customization tasks, use this option.
```

Figure 2-32   WebSphere Portal customization dialog - generate confirmation jobs for this customization task only

Enter a valid job card at the bottom of the panel shown in Figure 2-33 on page 70. Use a /*JOBPARM card in your jobs to ensure that they run on the correct LPAR.

It is not necessary to code a job card on the Portal ISPF panel where you generate jobs, because one will always be added for you. However, you have to edit the /*JOBPARM=* that is generated in order to add the correct LPAR name. (Do not include the USER or PASSWORD parameters in your job card, as they will be added automatically to the jobs that require them.)

When done, press Enter and the jobs will be created in the dataset specified in this panel. Press Enter again after the generation completes.



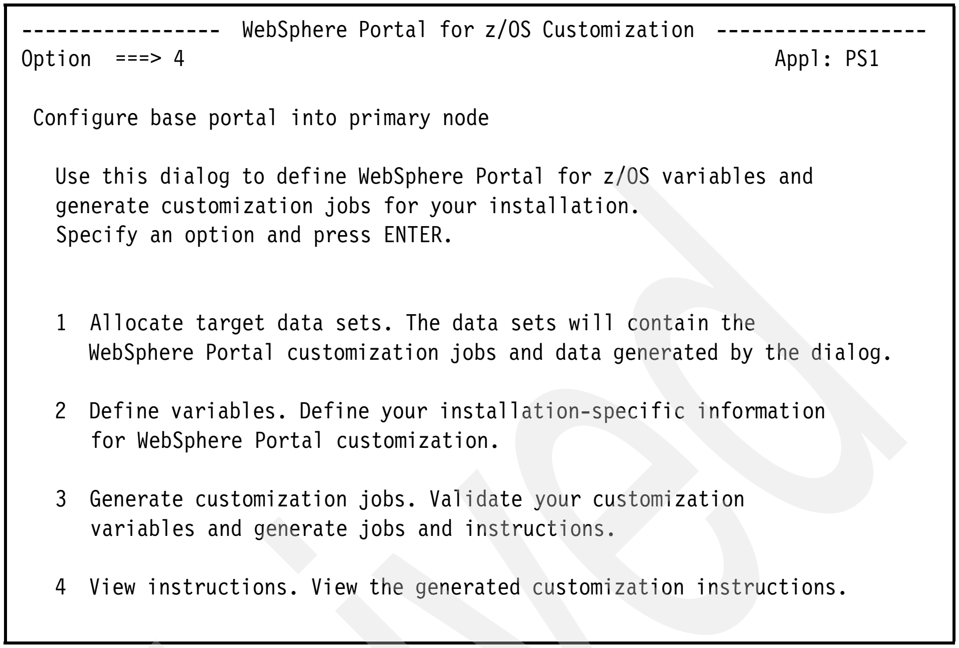
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'WPCELL.PORTALA.CNTL'
   'WPCELL.PORTALA.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.

 All the jobs that will be tailored for you will need a job card.
 Please enter a valid job card for your installation below. The
 file tailoring process will update the job name for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations.

 Specify the job cards, then press Enter to continue.

//jobname JOB (999,POK),'VANAERS',CLASS=A,REGION=0M,MSGCLASS=H,
//  NOTIFY=&SYSUID
//*
```

Figure 2-33 WebSphere Portal customization dialog - generate installation jobs

Note: Portal inserts a /*JOBPAR SYSAFF=* that will take precedence over any /*JOBPAR that may have been put into this panel.

Edit the /*JOBPARM in the jobs after they have been generated (for example, we used /*JOBPARM S=SC49,L=9999 in our jobs).

You can view instructions on running all the jobs that were generated by either choosing option **4** in the panel shown in Figure 2-34 on page 71, or by browsing the EJPIND1 member of your CNTL data set.



```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===> 4                                                    Appl: PS1

 Configure base portal into primary node

   Use this dialog to define WebSphere Portal for z/OS variables and
   generate customization jobs for your installation.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.
```

Figure 2-34   WebSphere Portal customization dialog - view job instructions

2.4 Preparing and running WebSphere Portal customization jobs

The next steps involve preparing and running the WebSphere Portal customization jobs. Before running the jobs, perform the tasks described in 2.4.1, "Tasks to be completed before running the customization jobs" on page 71.

2.4.1 Tasks to be completed before running the customization jobs

As mentioned, before you run the customization jobs, complete these tasks. Many of these tasks are also documented in HLQ.CNTL(EJPIND1).

1. Copy the current WebSphere Application Server PROCLIB member WPASRAZ to a new member of SYS1.PROCLIB. (Our member is called WPASRAP.)

   The PROC looks as follows. Notice the STEPLIB with the WebSphere Application Server and Portal Server datasets. WebSphere Portal dataset HLQ.SEJPLPA must be in the STEPLIB of this procedure if it is not in LINKLST.

```
//* Output DDs
//*
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//*
//*Steplib Setup
//*
//STEPLIB  DD DISP=SHR,DSN=BBWP6049.SBBOLD2
//         DD DISP=SHR,DSN=BBWP6049.SBBOLOAD
//         DD DISP=SHR,DSN=BBWP6049.SEJPLPA
```

2. APF-authorize HLQ.SEJPLPA. Keep in mind that *all* libraries in a concatenation must be authorized; otherwise, the authorization is lost.
3. Update your active BPXPRMxx member to have the WebSphere Portal for z/OS configuration HFS (for example, MVS.WAS6.WPCELL.PORTALA.CONFIG.HFS) mounted at the WebSphere Portal home directory mount point (/waswpconfig/wpcell/wpport) in read/write mode. You can do this as follows:

   ```
   MOUNT FILESYSTEM('OMVS.WAS6.WPCELL.PORTALA.CONFIG.HFS')
     MOUNTPOINT('/waswpconfig/wpcell/wpport')
       TYPE(ZFS)
       NOAUTOMOVE
       SYSNAME(SC49)
       PARM(‘AGGRGROW’)
       MODE(RDWR)Jobs.
   ```

   Repeat this step for all HFS file systems you use.

2.4.2 Running the WebSphere Portal customization jobs

The WebSphere Portal customization jobs must be executed in the sequence listed here.

1. EJPSRACF

   This job creates the keyring and certificates required for SSL. It also creates the started task profiles for the WebSphere Portal control and servant regions. Ensure that this job is run with an ID that has the authority in RACF to create these items.

> **Note:** The RACF commands that are issued by EJPSRACF have already been issued by earlier jobs during the creation of the Deployment Manager and Empty node. Check the commands being issued in EJPSRACF and if the profiles and keyring already exist, do not submit EJPSRACF.

2. EJPSCHFS

   This job creates the mount point directory for WebSphere Portal's configuration HFS, and mounts the data set to the mount point. Ensure that this job is run with an ID that has a UID of 0, so that it has sufficient authority to create files and directories.

   This ID must also have an OMVS segment so that it can write the installation log to this directory structure. Set the permissions of the WebSphere Portal mount point to 775 so that the WebSphere Application Server administrative user and group can write to this directory structure.

   (As mentioned, in our case we used symbolic links to address the HFS contents; refer to Figure 2-33 on page 70.)

Example 2-4 EJPSCHFS directory creation

```
Display File Attributes

Pathname : /waswpconfig/wpcell/wpport
                                     More:     +
File type . . . . . . : Directory
Permissions . . . . . : 775 rwxrwxr-x
Access control list . : 0
File size . . . . . . : 256
File owner  . . . . . : WPADMIN(8000)
Group owner . . . . . : WPCFG(8500)
Last modified . . . . : 2007-03-01 14:00:06
Last changed  . . . . : 2007-03-01 14:00:10
Last accessed . . . . : 2007-03-01 14:00:06
Created . . . . . . . : 2007-03-01 14:00:06
Link count  . . . . . : 2
```

3. EJPSCFG1

   If using a Network Deployment environment, start the Deployment Manager and Node Agent before running this job. This job performs an archive install of the WebSphere Portal configuration files. In order to complete this, the WebSphere Portal install program needs access to a zip and unzip utility. Free versions of the zip and unzip utilities can be found at the following site:

   `http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty1.html`

   After the utilities are downloaded, the .profile of the user running the job must point to the location of the utilities so that the job knows where to find them.

The line you add to .profile should look like this:

```
export PATH=$PATH:/ziptool/utility/path/
```

4. EJPSNDC1

   This job configures a base portal into the primary node of the Network Deployment cell. If the WebSphere Portal application server is named something other than WebSphere_Portal, the name of the server will have to be changed from WebSphere_Portal to the custom name[1]. This is done in the deploy.properties file located in each of the subdirectories listed in Table 2-2. Each of these subdirectories can be found by navigating to:

   ```
   /waspconfig/wpcell/wpport/Portal/config/deployable/ear/
   ```

   The Application Name field is the name given to the application in the WebSphere administrative console under **Applications** -> **Enterprise Applications**. These subdirectories are utilized by the applications listed in the left column. There are eight deploy.properties files that need to be updated.

   The deploy.properties files are in ASCII, so an ASCII editor is required to change them. Some of the deploy.properties files also contain the WebSphere Portal Administrator user ID/password. If a custom Portal administrator user ID is being utilized, it will have to be corrected in these files as well. The default ID is wpsadmin.

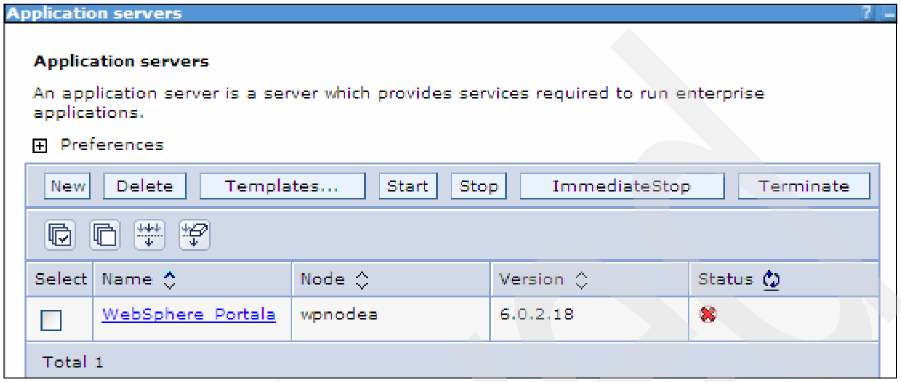
EJPSNDC1 takes some time to run. The output can be viewed while the job is running by viewing tmp/ejp.ndcxx.out, where xx is from 11 to 15 depending on the phase the job is in.

Table 2-2 The ear files that are updated if using a custom application server name

Application name	Subdirectory name
LWP_Scheduler_Resource	admin.wpsched.svc.ear.prod
LWP_CAI	cai.svc.ear.prod
LWP_Security_Ext	sec.ext.svc.ear.prod
LWP_TAI	tai.svc.ear.prod
LWP_Mail_Servlets	cpp.mail.ear.prod
LWP_People	people.impl.ear.prod
WSPolicyManager	wp.policy.webservices.ear.prod
TemplateLibrary_Servlets	wp.ap.app.templates.ear.prod

[1] At the time of writing, this likely to be changed in an upcoming PTF.

Some parameters can verify that job EJPSNDC1 ran correctly. At the WebSphere Application Server administrative console, select **Servers** -> **Application Servers**. You should find an application server defined with the name as specified during the customization process; see Figure 2-35.



Application servers

Application servers

An application server is a server which provides services required to run enterprise applications.

⊞ Preferences

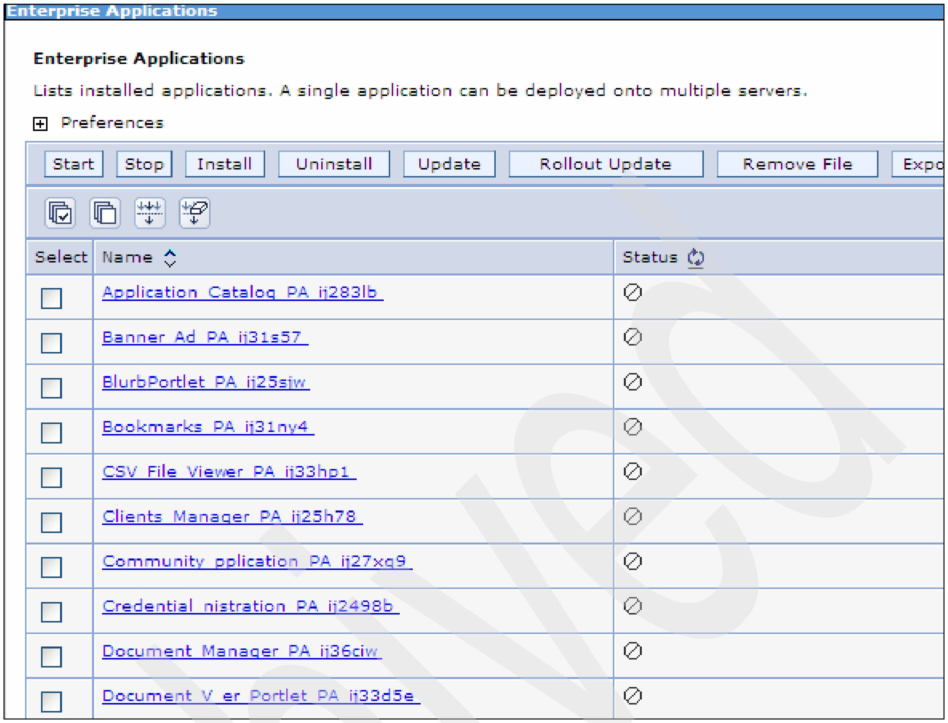
Select	Name	Node	Version	Status
□	WebSphere_Portala	wpnodea	6.0.2.18	✗

Total 1

Figure 2-35 Verify the application server was defined

In addition to defining an application server, EJPSNDC1 also deployed applications required and used by WebSphere Portal. To verify this, at the WebSphere administrative console, select **Applications** -> **Enterprise Applications**.

The output should be as shown in Figure 2-36 on page 76. There should be 105 applications listed.



Enterprise Applications

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

⊞ Preferences

Select	Name	Status
□	Application_Catalog_PA_ij283lb	⊘
□	Banner_Ad_PA_ij31s57	⊘
□	BlurbPortlet_PA_ij25sjw	⊘
□	Bookmarks_PA_ij31ny4	⊘
□	CSV_File_Viewer_PA_ij33hp1	⊘
□	Clients_Manager_PA_ij25h78	⊘
□	Community_pplication_PA_ij27xq9	⊘
□	Credential_nistration_PA_ij2498b	⊘
□	Document_Manager_PA_ij36ciw	⊘
□	Document_V_er_Portlet_PA_ij33d5e	⊘

Figure 2-36 Verify that applications were deployed

2.4.3 Making changes for WebSphere Portal in WebSphere Application Server

Before the WebSphere Portal application server can be started, you must complete a few other steps:

1. Verify that all the port numbers used by the WebSphere Portal application server are unique.

   Appendix B, “Additional material” on page 457 includes a jython script that will change all the ports used by WebSphere Portal to a specific range of numbers.

   These changes can also be made via the WebSphere administrative console by selecting **Servers** -> **Application Servers** -> **WebSphere_Portal** -> **Communication** -> **Ports**.

   All HTTP/HTTPS ports should also be changed in the default virtual host by selecting **Environment** -> **Virtual Hosts** -> **default_host** -> **Host Aliases**.

Finally, update the HTTP transport port numbers for the WebSphere Portal application server by selecting **Servers** -> **Application Servers** -> **WebSphere_Portal** -> **Container Settings** -> **Web Container Settings** -> **HTTP Transports**.

The ports in {Portal_Home}/config/wpconfig.properties should match the ports defined to WebSphere Application Server for the WebSphere Portal application server.

Here is an example of the ports defined in wpconfig.properties that must match their WebSphere Application Server counterparts. Also, the XmlAccessPort must be the same value as the WpsHostPort in wpconfig.properties.

- WpsHostPort=29818
- WpsSoapPort=29810
- XmlAccessPort=29818
- DmgrSoapPort=29910

2. If the WebSphere Portal data set HLQ.SEJPLPA is not in the linklist, update the WebSphere Portal started task procedure to point to the member where this data set was added to STEPLIB. The Z= value should be the member in SYS1.PROCLIB.

   ```
   //WPASRA  PROC ENV=,Z=WPASRAP
   ```

   Keep in mind that WebSphere Portal uses the same started task procedure as WebSphere Application Server. A new SYS1.PROCLIB member is not required.

   The start command for the WebSphere Portal address space refers to the WebSphere Application Server procedures, and uses a WebSphere Portal environment which points to the appropriate was.env file (a file in EBCDIC encoding that is the result of a compilation of the xml files describing this particular server).

3. Change the WCM_PORT at the server level in the WebSphere administrative console. Go to **Environment** -> **WebSphere Variables** and search for the WCM_PORT.

   WCM_PORT is found at the application server level. This port should be set to the same port as the portal http port (in our case, 29818).

- Start the WebSphere Portal application server using the WebSphere administrative console; select **Servers** -> **Application Servers**.

  Select the check box next to the WebSphere_Portal server and click **Start**.

  Verify that the Portal is starting correctly by going into the PORTAL address space in SDSF. An INITIALIZATION COMPLETE message should be displayed in both the control and servant address spaces, as follows:

```
+BBOOO020I INITIALIZATION COMPLETE FOR WebSphere FOR Z/OS SERVANT
PROCESS PORTALA
```

2.4.4 Running the last jobs

At this point, there are two jobs left to run before you go to the browser to access the Portal; run these jobs:

- EJPSPCHK

  This job verifies the installation of WebSphere Portal by making sure the system is up and checking to see whether it can accept input. EJPSPCHK should be run with your WebSphere Administration Server administration ID.

  If this job fails, there are a few areas to check. The ID running the job must have access to the Java directories. Add a line to the .profile of the WebSphere Administration Server administration ID that points to Java. Here is an example that can be customized to point to any specific Java path:

  ```
  export JAVA_HOME=/usr/lpp/java/J1.4
  ```

  Problems with this job can also occur if the domain name is not correct in wpconfig.properties. The last thing to check are the ports defined in wpconfig.properties. WpsHostPort must match XmlAccessPort; otherwise, EJPSPCHK will fail in the first step with a return code of 256.

- EJPSPACT

  This job activates all the portlets within WebSphere Application Server. It is primarily used for verification.

2.5 Accessing your Portal

After all the jobs have been run, go to the WebSphere Portal URL:

`http://hostname:port/wps/portal`

In our scenario, the URL is:

`http://wtsc49.itso.ibm.com:29818/wps/portal`

Log on with the WebSphere Portal administrator ID and password that were defined in the install dialog. The logon window is shown in Figure 2-37 on page 79.

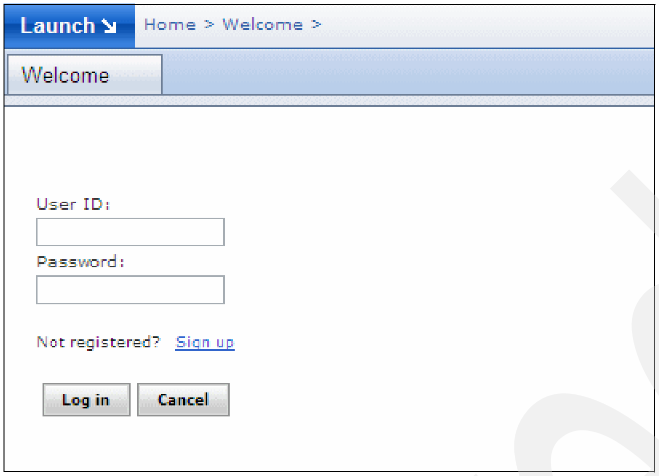
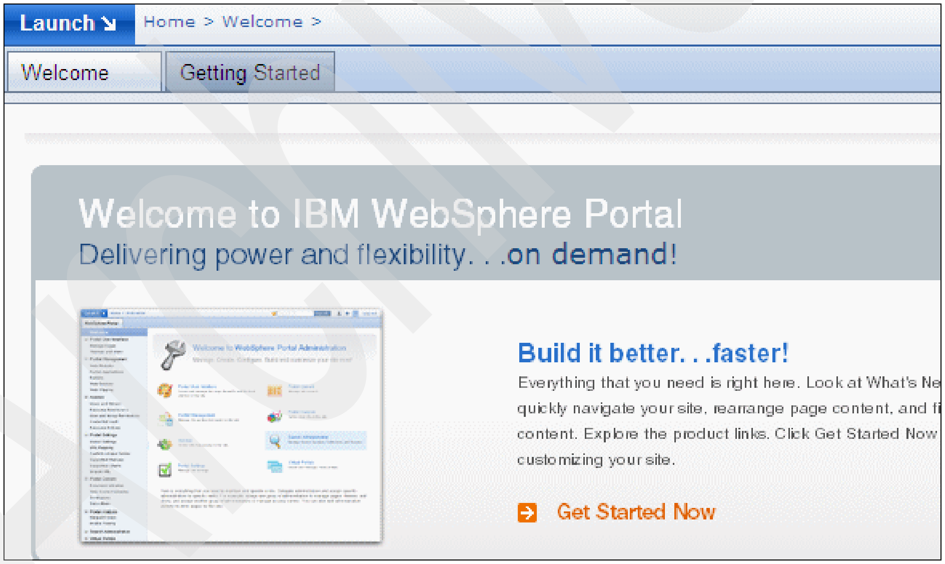


Figure 2-37 Logon page: http://wtsc49.itso.ibm.com:29818/wps/portal

After you log on, a welcome screen similar to Figure 2-38 will appear.



Welcome to IBM WebSphere Portal
Delivering power and flexibility. . .on demand!

Build it better. . .faster!

Everything that you need is right here. Look at What's Ne
quickly navigate your site, rearrange page content, and fi
content. Explore the product links. Click Get Started Now
customizing your site.

Figure 2-38 WebSphere Portal Server welcome screen

2.6 Creating a Web server definition and generating the HTTP plug-in

The last task we completed in the basic configuration of our environment was to set up an HTTP server to communicate with WebSphere Application Server. We set up the HTTP server so that the WebSphere Portal URL can be accessed through the default HTTP port (port 80) instead of having to enter the WebSphere Portal virtual host in the URL. Our HTTP server was installed on a z/Linux partition of a VM LPAR.

Although this remote Web server cannot be managed via the WebSphere administrative console, we must still define it in order to generate the WebSphere plug-in file. This process is started by going into the WebSphere administrative console and selecting **Servers** -> **Web servers**.

- Click **New** to create a new Web server.
- At this point there is only one node to choose on the "Step 1: Select a node" screen. In Chapter 6, "Implementing high availability with WebSphere Portal on z/OS" on page 165, we create a second node that a Web server can be defined to.

  Give the new Web server a name in the Server name field. (This name has no connection to the actual HTTP server name; it can be anything.)
- The next screen that appears is "Step 2: Enter the properties for the new Web server". "Type" is the Web server software that is being used. We are using IBM HTTP Server (IHS).

  Port 80 is the default HTTP port that a browser looks for. If something other than 80 is used, it will have to be entered into the URL.

  The final required field is Plug-in installation location. This is where the plug-in code is installed.
- There is only one template available in "Step 3: Select a Web server template" called IHSZOS. Select the template and click **Next**.
- Step 4 asks for confirmation of the values chosen in the previous steps; click **Finish**.
- Save the changes in the WebSphere administrative console and recycle the Deployment Manager.

After a Web server is defined, it must be added to the target mappings of every WebSphere Portal application so that each WebSphere Portal application is entered into the plugin-cfg.xml file, and so that the plugin-cfg.xml knows about the virtual hosts that are defined for WebSphere Portal.

Normally for an application that only has one ear file, you would add the new Web server to the target mappings in the WebSphere administrative console. Here, however, go to **Applications** -> **Enterprise Applications** -> ***Application_Name*** **> Map modules to servers**. Select the application server and the Web server that are being mapped to, and then click **Apply**.

WebSphere Portal is composed of a number of applications that are installed into WebSphere Application Server. Updating each of these ear files through the WebSphere administrative console would be very time-consuming. A JACL script has been created to make this task easier.

The JACL script maps an application server or Web server that is specified to all of the WebSphere Portal applications. This script is included in Appendix B, “Additional material” on page 457. After the script has completed, regenerate the plug-in.

Alternatively, you can generate a plug-in by using the GenPluginCfg.sh script. See the WebSphere for z/OS Infocenter article at the following site:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.WebSphere.zseries.doc/info/zseries/ae/rxml_genplugincfg1.html`

In summary, the process is as follows:

1. Open an OMVS session and navigate to the Deployment Manager's ../profile/default/bin directory.
2. Run the GenPluginCfg.sh without any parameters. A plugin-cfg.xml file is generated to the Deployment Manager's ../profile/default/config/cells directory. It will contain all information about the current WebSphere configuration and all the Portal applications.

   Later, if you configure a cluster, re-run GenPluginCfg.sh after creating the cluster so that plugin-cfg.xml contains the definitions of the cluster's topology.

Viewing the plug-in now will display the WebSphere Portal information. Example 2-5 includes sample sections of the plugin-cfg.xml file for our system. (The dotted lines represent lines of the plug-in that have been removed.)

This first section is a list of virtual hosts. Virtual host 29818 and 29819 are the ports used for WebSphere Portal.

The second section starts with the server definition. The `Name` parameter includes the node name and the application server name.

Within this server, there are two transports defined. One transport is for http and the other for https. Each transport includes the hostname of the portal application server and the port used for the protocol.

Example 2-5 Sample lines from plugin-cfg.xml

```
.........................................
<VirtualHostGroup Name="default_host">
      <VirtualHost Name="*:29918"/>
      <VirtualHost Name="*:29919"/>
      <VirtualHost Name="*:80"/>
      <VirtualHost Name="*:29818"/>
      <VirtualHost Name="*:29819"/>
   </VirtualHostGroup>
.........................................
<Server ConnectTimeout="0" ExtendedHandshake="false" MaxConnections="-1"
Name="ndnodea_WebSphere_Portala" ServerIOTimeout="0" WaitForContinue="false">
         <Transport Hostname="wtsc49.itso.ibm.com" Port="29818" Protocol="http"/>
         <Transport Hostname="wtsc49.itso.ibm.com" Port="29819" Protocol="https">
.........................................
<UriGroup Name="default_host_WebSphere_Portala_ndnodea_Cluster_URIs">
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/wps/PA_npbyd19/*"/>
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/wps/PA_mtfl6ck/*" />
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/wps/PA_mtetmjg/*" />
104 more applications listed........
```

2.7 Conclusion

After completing activities described in this chapter, you will have a working WebSphere Portal using the Cloudscape database. However, there is no real security at this point, other than having defined the necessary RACF profiles to run the server.

In Chapter 3, “Migrating the Portal database from Cloudscape to DB2” on page 83, we explain how to move your database from Cloudscape to DB2.

3

Migrating the Portal database from Cloudscape to DB2

WebSphere Portal is shipped with a Cloudscape database to use for its back-end data. Although Cloudscape is sufficient to get WebSphere Portal up and running, it is desirable to use a more robust database when moving the Portal into a production environment. This chapter describes the process of moving all of the back-end data used by WebSphere Portal over to a DB2 database.

We also discuss here another reason to migrate to DB2, which is the intention to use clustered Portal servers. These require shared data management or any configuration with multiple servants.

3.1 DB2 as a data repository

WebSphere Portal uses a database to store all of its configuration, access control, and user data. By default, WebSphere Portal is installed into an open source Cloudscape database that comes with the product.

Cloudscape, however, has a few limitations which make it insufficient for use in a production environment. Cloudscape is a single-threaded database. If a thread hangs in Cloudscape, the Portal will become unavailable.

Also, Cloudscape does not yet support horizontal cloning, clustered environments, or enabling security in a database-only mode. Using an external database, such as DB2, provides all the required functionality, performance and recoverability that is inherent with the product.

This chapter discusses all the steps necessary to move the data that is stored in the limited Cloudscape database over to a robust DB2 database.

3.2 Configuring DB2 as a database for WebSphere Portal

The preparation required to switch database repositories is similar to that of the WebSphere Portal software install. Batch jobs are generated based on parameters specified in the ISPF install dialogs. Before running the install dialog, the ID that will run the jobs must have authority in DB2 to create databases.

3.2.1 Setting up JDBC to access DB2

WebSphere Portal accesses DB2 through a Java DataBase Connectivity™ (JDBC) connection. JDBC is implemented through drivers which are provided by the database vendor. DB2 is shipped with two types of drivers:

- A JDBC Type 2 driver for local DB2 connections
- A JDBC Type 4 driver for remote DB2 connections using DDF in DB2

We use the JDBC Type 2 driver. This requires:

1. A JDBC datasource definition under the Type 2 universal driver. This is created by one of the batch jobs that is generated through the ISPF dialog.
2. An DB2JccConfiguration.properties member which has to be prepared by us in the /etc directory in UNIX Systems Services. In our scenario we have a file called /etc/D8J1/DB2JccConfiguration.properties.

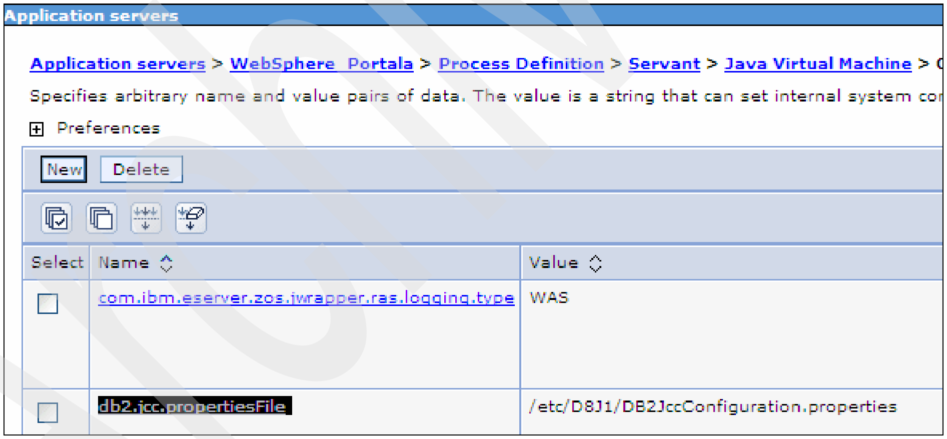
The DB2JccConfiguration.properties file must be created before running the ISPF dialog. This file contains one line:

```
db2.jcc.ssid=D8J1
```

`D8J1` is the name of our DB2 subsystem where the WebSphere Portal databases are created. We set the permissions of the file to 775 so that anyone can read it. (Otherwise, some of the batch jobs will fail.) The location of the DB2JccConfiguration.properties file is entered into the ISPF dialog.

> **Note:** The DB2JccConfiguration.properties file is set as a custom property of the servant region Java Virtual Machine (JVM) by the main database transfer job EJPSDBT. This property points to the properties file that is created in step 2.

After the application server is defined, this property can be found in the WebSphere administrative console; see Figure 3-1. You can locate the db2.jcc.propertiesFile property by selecting **Servers** → **Application servers** → **WebSphere_Portala** → **Process Definition** → **Servant** → **Java Virtual Machine** → **Custom Properties**.



Application servers

Application servers > WebSphere_Portala > Process Definition > Servant > Java Virtual Machine > C

Specifies arbitrary name and value pairs of data. The value is a string that can set internal system co

⊞ Preferences

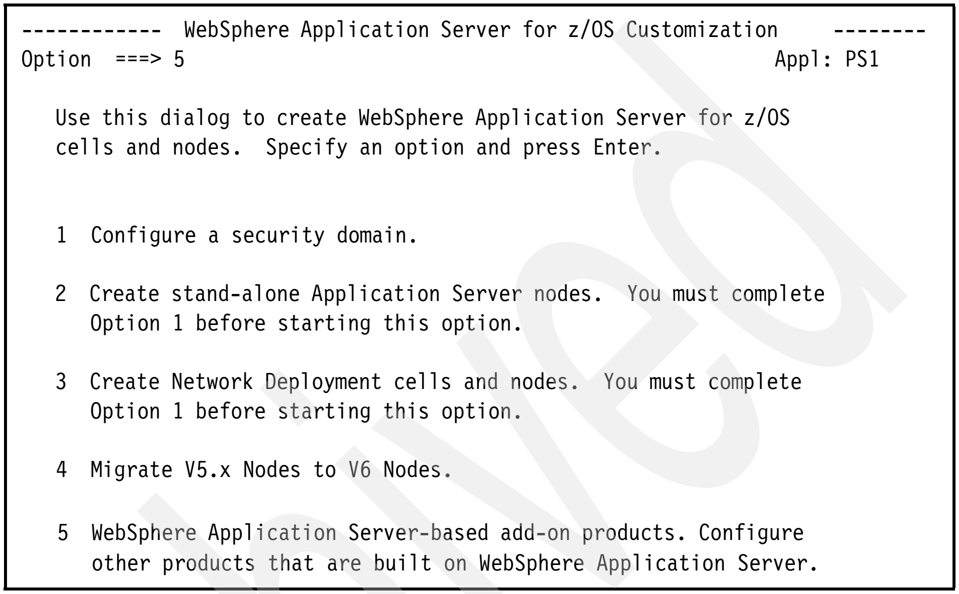
Select	Name	Value
□	com.ibm.eserver.zos.jwrapper.ras.logging.type	WAS
□	db2.jcc.propertiesFile	/etc/D8J1/DB2JccConfiguration.properties

Figure 3-1 WebSphere admin console - JDBC properties file

Next we executed the WebSphere install dialog. This is the same dialog we launched from ISPF option 6 in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35. (Many of the panels may look familiar because we navigated through them in that chapter.)

3.2.2 Using the ISPF customization panels

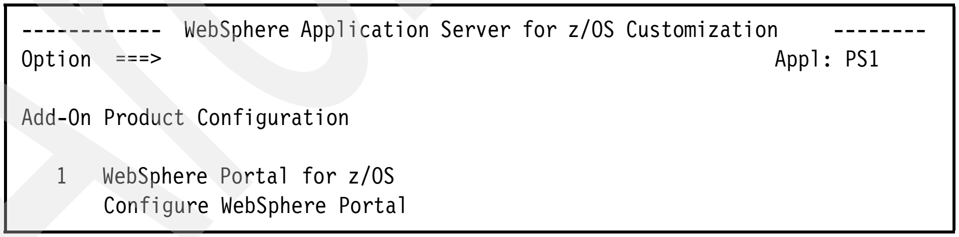
On the main panel, we selected option **5** for WebSphere Application Server add-on products, as shown in Figure 3-2.



```
-----------  WebSphere Application Server for z/OS Customization    --------
Option  ===> 5                                                  Appl: PS1

   Use this dialog to create WebSphere Application Server for z/OS
   cells and nodes.  Specify an option and press Enter.


   1  Configure a security domain.

   2  Create stand-alone Application Server nodes.  You must complete
      Option 1 before starting this option.

   3  Create Network Deployment cells and nodes.  You must complete
      Option 1 before starting this option.

   4  Migrate V5.x Nodes to V6 Nodes.

   5  WebSphere Application Server-based add-on products. Configure
      other products that are built on WebSphere Application Server.
```

Figure 3-2 WebSphere Application Server for z/OS Customization panel - selecting WebSphere Application Server add-on products

We chose the option for WebSphere Portal for z/OS, as shown in Figure 3-3.

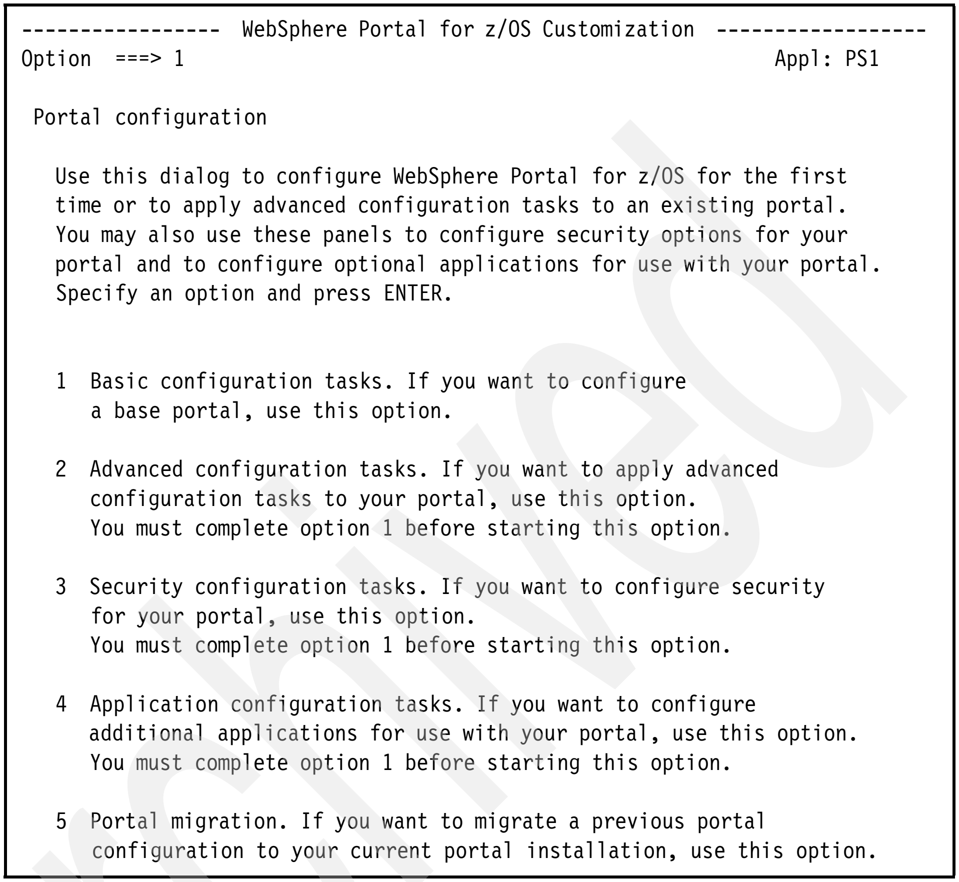


```
-----------  WebSphere Application Server for z/OS Customization    --------
Option  ===>                                                    Appl: PS1

Add-On Product Configuration

   1   WebSphere Portal for z/OS
       Configure WebSphere Portal
```

Figure 3-3 Customization panel - selecting WebSphere Portal

The next panel will show the version; we pressed Enter and bypassed this screen.

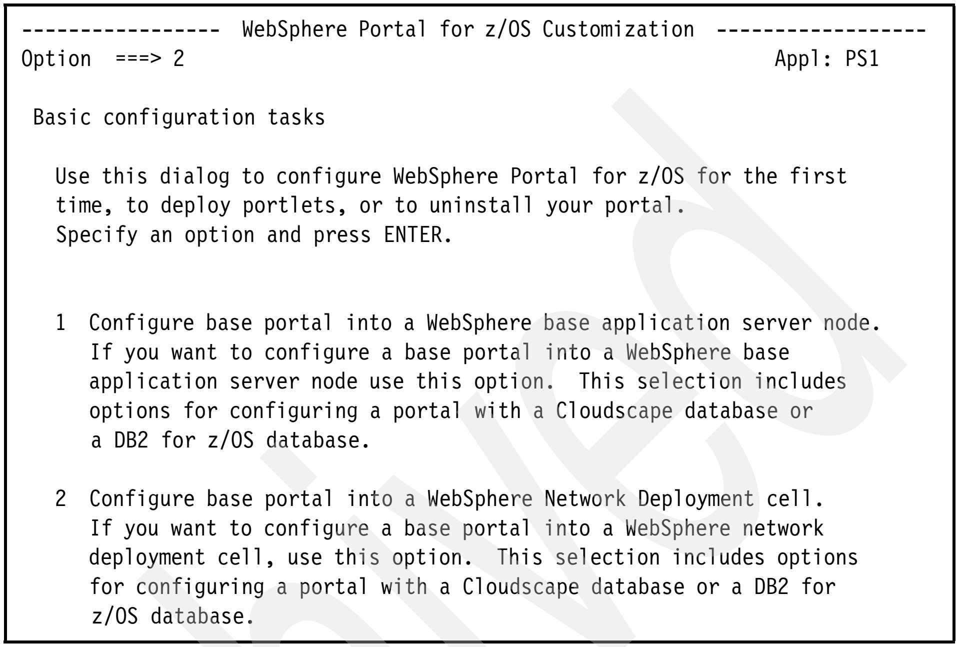
In the panel shown in Figure 3-4, we selected **1** for base configuration tasks.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                    Appl: PS1

 Portal configuration

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time or to apply advanced configuration tasks to an existing portal.
   You may also use these panels to configure security options for your
   portal and to configure optional applications for use with your portal.
   Specify an option and press ENTER.


   1  Basic configuration tasks. If you want to configure
      a base portal, use this option.

   2  Advanced configuration tasks. If you want to apply advanced
      configuration tasks to your portal, use this option.
      You must complete option 1 before starting this option.

   3  Security configuration tasks. If you want to configure security
      for your portal, use this option.
      You must complete option 1 before starting this option.

   4  Application configuration tasks. If you want to configure
      additional applications for use with your portal, use this option.
      You must complete option 1 before starting this option.

   5  Portal migration. If you want to migrate a previous portal
      configuration to your current portal installation, use this option.
```

Figure 3-4   Customization panel - basic configuration tasks

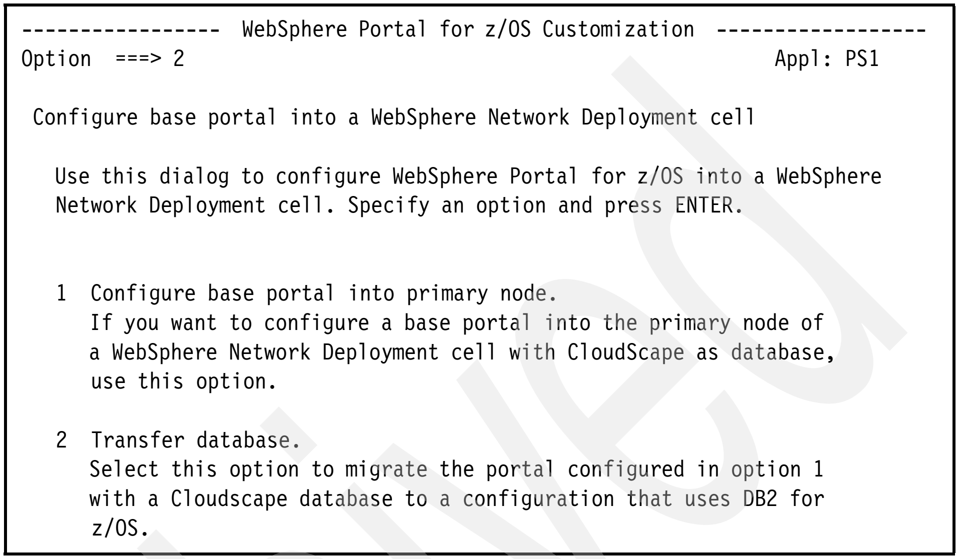
In the panel shown in Figure 3-5, you select option 1 if a base application server node is used. Select option 2 if Network Deployment is installed.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                  Appl: PS1

 Basic configuration tasks

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time, to deploy portlets, or to uninstall your portal.
   Specify an option and press ENTER.


   1  Configure base portal into a WebSphere base application server node.
      If you want to configure a base portal into a WebSphere base
      application server node use this option.  This selection includes
      options for configuring a portal with a Cloudscape database or
      a DB2 for z/OS database.

   2  Configure base portal into a WebSphere Network Deployment cell.
      If you want to configure a base portal into a WebSphere network
      deployment cell, use this option.  This selection includes options
      for configuring a portal with a Cloudscape database or a DB2 for
      z/OS database.
```

Figure 3-5 Customization panel - select base application server node or Network Deployment

This is the first option that differs from Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35. We chose option **2** to proceed with creating the database transfer jobs, as shown in Figure 3-6.



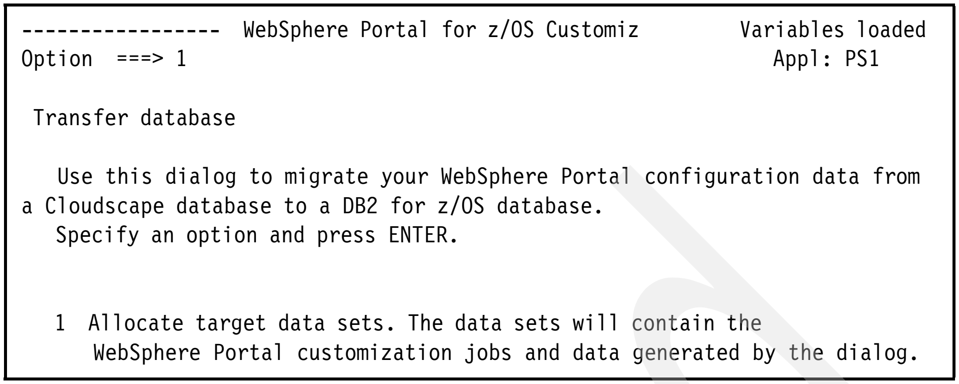
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                  Appl: PS1

 Configure base portal into a WebSphere Network Deployment cell

   Use this dialog to configure WebSphere Portal for z/OS into a WebSphere
   Network Deployment cell. Specify an option and press ENTER.


   1  Configure base portal into primary node.
      If you want to configure a base portal into the primary node of
      a WebSphere Network Deployment cell with CloudScape as database,
      use this option.

   2  Transfer database.
      Select this option to migrate the portal configured in option 1
      with a Cloudscape database to a configuration that uses DB2 for
      z/OS.
```

Figure 3-6 Customization panel - select database transfer

It will make the rest of the customization easier if custom variables are loaded at this time. Loading these variables will put the values that were chosen during the install into the database transfer panels where required.

In our case, we chose option **L** to load the variables. After they were loaded, we chose option **1** to create new data sets to store the database transfer JCL and data.

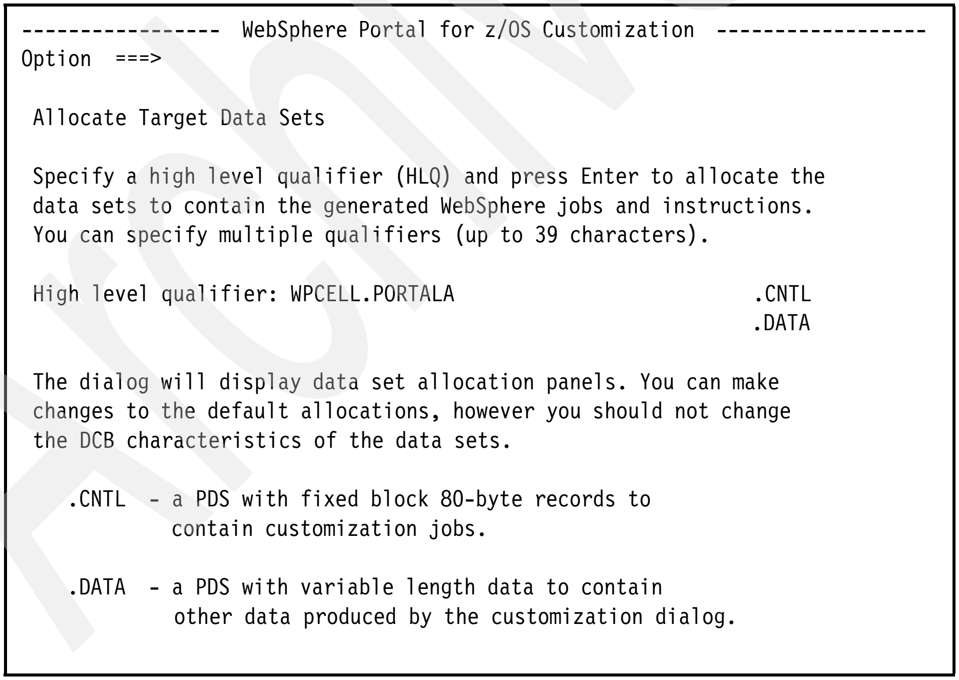
> **Note:** If this step is skipped, the jobs will be stored in the data set defined for the WebSphere Portal install, which was loaded with the custom variables.
>
> It is good practice to choose option 1 in order to verify the data set being used any time the option is presented.



```
-----------------  WebSphere Portal for z/OS Customiz        Variables loaded
Option  ===> 1                                                  Appl: PS1

 Transfer database

   Use this dialog to migrate your WebSphere Portal configuration data from
a Cloudscape database to a DB2 for z/OS database.
   Specify an option and press ENTER.


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

Figure 3-7 Customization panel - allocate data sets

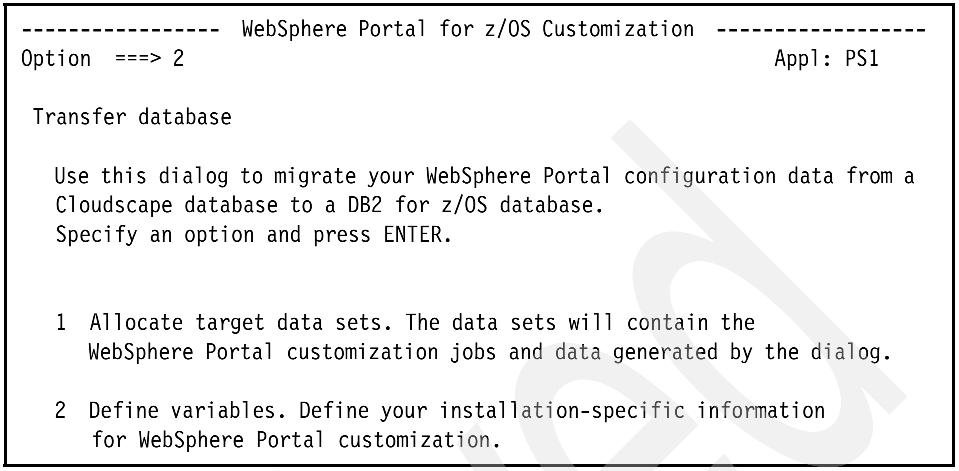
In the panel shown in Figure 3-8, you can change the data set name to store the generated jobs into a different data set than the WebSphere Portal install jobs. In our scenario, however, we kept all the install jobs in the same data set for ease of accessibility.



```
-----------------  WebSphere Portal 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: WPCELL.PORTALA                        .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 customization jobs.

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

Figure 3-8 Customization panel - change the data set name

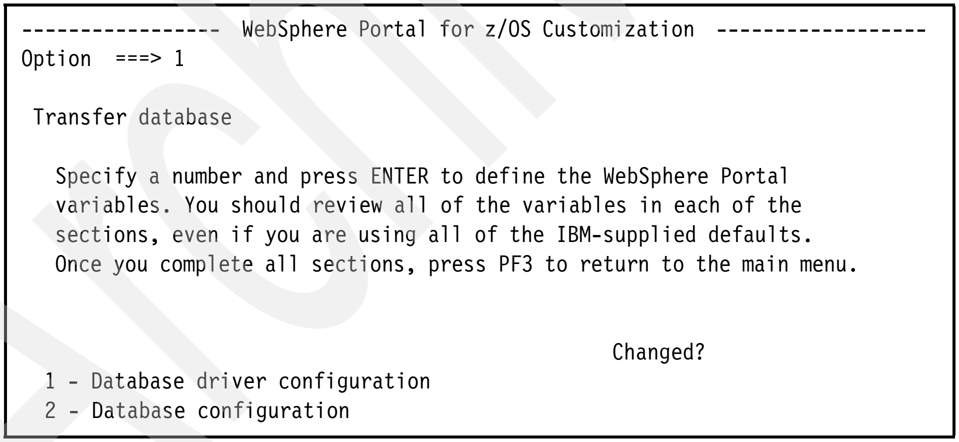
In the panel shown in Figure 3-9, we chose option **2** to define the variables.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                  Appl: PS1

 Transfer database

   Use this dialog to migrate your WebSphere Portal configuration data from a
   Cloudscape database to a DB2 for z/OS database.
   Specify an option and press ENTER.


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

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

Figure 3-9 Customization panel - define variables

In the panel shown in Figure 3-10, we selected option **1** to specify database driver information.

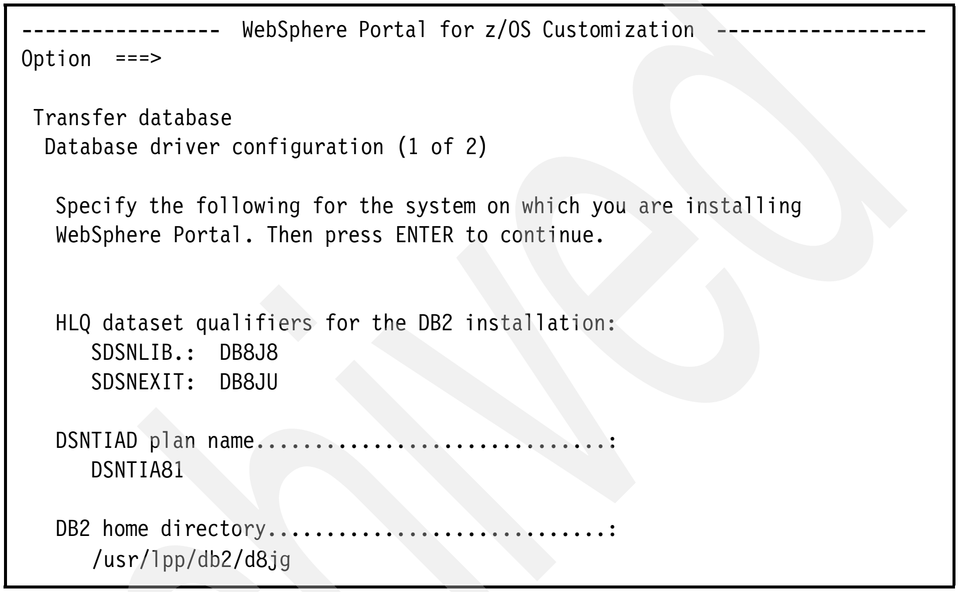


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1

 Transfer database

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Database driver configuration
  2 - Database configuration
```

Figure 3-10 Customization panel - select database driver configuration

In the panel shown in Figure 3-11, the SDSNLIB and SDSNEXIT fields are where the full high level qualifiers (HLQs) for the DB2 SDSNLOAD and SDSNEXIT data sets are specified.

The DSNTIAD plan usually has this name, but some systems use the version-specific name such as the one in the Figure 3-11. DB2 home directory is the HFS where DB2 is installed.



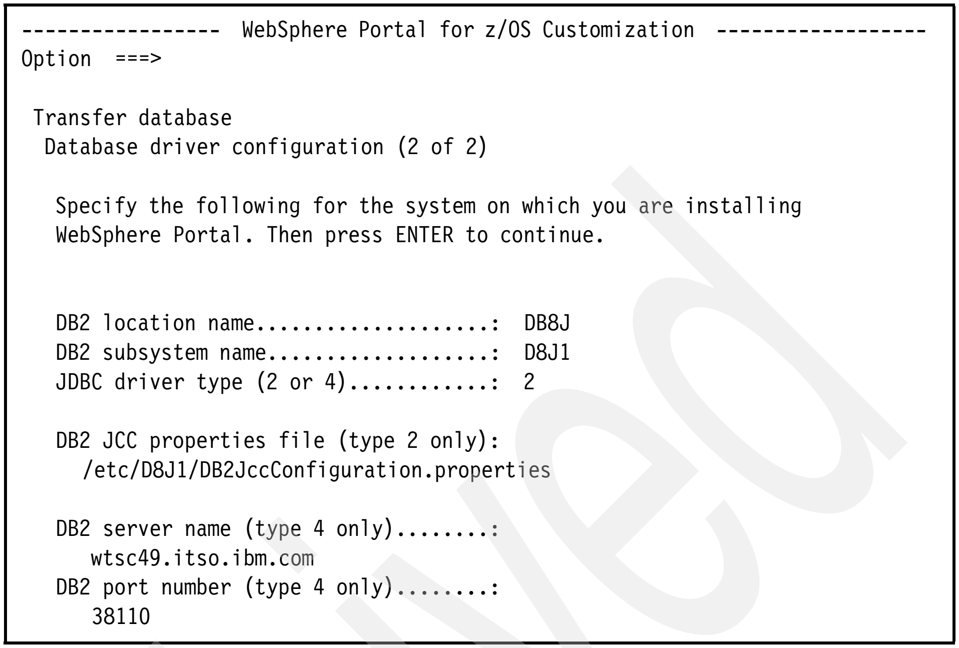
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Transfer database
  Database driver configuration (1 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   HLQ dataset qualifiers for the DB2 installation:
      SDSNLIB.:  DB8J8
      SDSNEXIT:  DB8JU

   DSNTIAD plan name..............................:
      DSNTIA81

   DB2 home directory.............................:
      /usr/lpp/db2/d8jg
```

Figure 3-11 Customization panel - DB2 variables

In the next panel, shown in Figure 3-12 on page 93, `DB2 location name` is the DDF location name of the DB2 subsystem. Issue `/recognitioncharacter(SRC) DIS DDF` to display your DB2 DDF information.

`DB2 subsystem name` is the 4-character subsystem name for the DB2 being used. Our scenario uses the Type 2 JDBC driver. Because we are using a Type 2 driver, we need to specify a JCC properties file. This is the file we created before launching the install dialog. The file must be named DB2JccConfiguration.properties, because any other name causes an error in the dialog. The directory path should be the path used when the properties file was created.

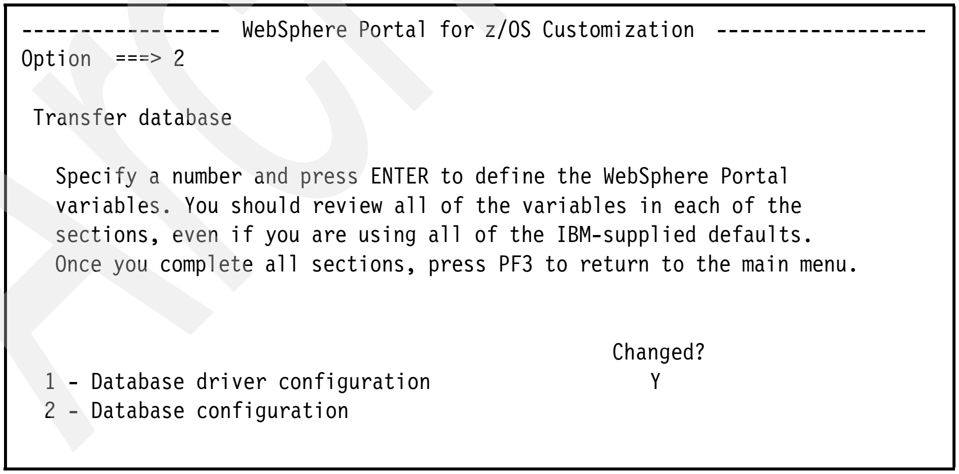
The `DB2 server name` and `DDF port number` have been included to fill out the panel. The DB2 port number must *never* be empty and can contain any numeric value. The database transfer job (EJPSDBT) requires the DB2 server name and DB2 port number.



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

 Transfer database
  Database driver configuration (2 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   DB2 location name....................:  DB8J
   DB2 subsystem name...................:  D8J1
   JDBC driver type (2 or 4)............:  2

   DB2 JCC properties file (type 2 only):
     /etc/D8J1/DB2JccConfiguration.properties

   DB2 server name (type 4 only)........:
      wtsc49.itso.ibm.com
   DB2 port number (type 4 only)........:
      38110
```

Figure 3-12   Customization panel - DB2 variables (continued)

In the panel shown in Figure 3-13, we chose **2** to enter the database configuration parameters.



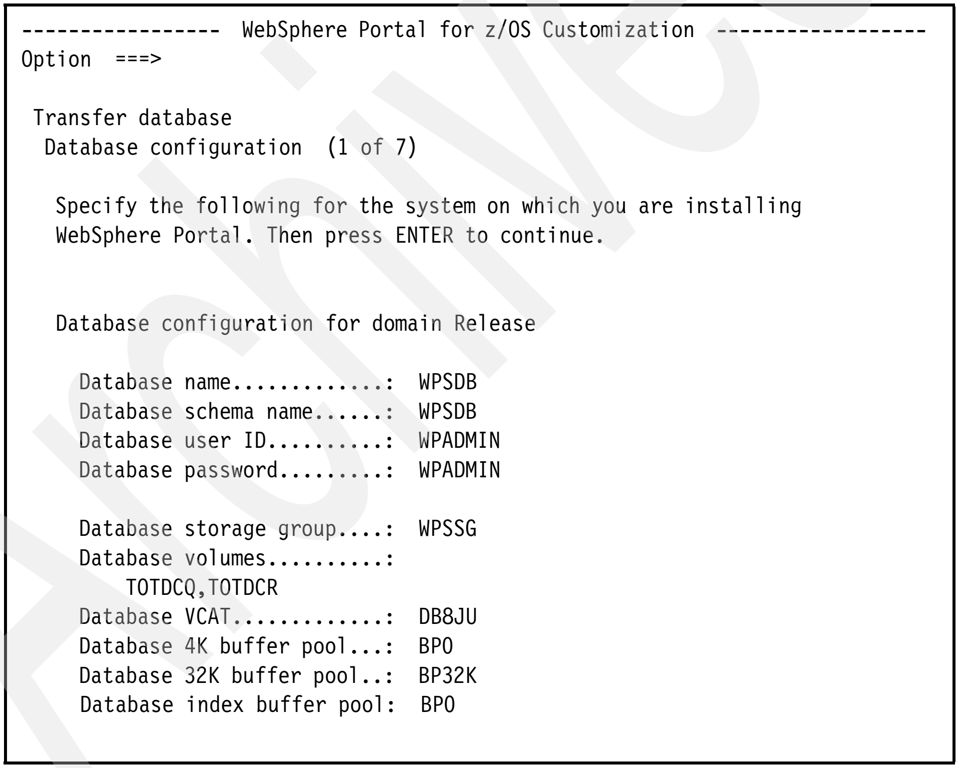
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2

 Transfer database

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Database driver configuration                 Y
  2 - Database configuration
```

Figure 3-13   Customization panel - database configuration

One of the jobs that are run during the database transfer process creates all the database objects in DB2 that are required for WebSphere Portal. Several databases are created by the job. Each database has a specific use by the WebSphere Portal product. In the next 7 panels, make sure each database name and schema is *unique* between the panels.

This panel is for the release database. The schema name can match the database name. The database user ID is the ID that has the authority required to create a database on our DB2 subsystem. The storage group specified gets created with the volumes entered. Enter a comma (,) between the database volume names to define the storage group with multiple volumes. If the system where the Portal is being installed uses SMS, put an asterisk (*) in this field. The database VCAT and buffer pools are all system-specific.

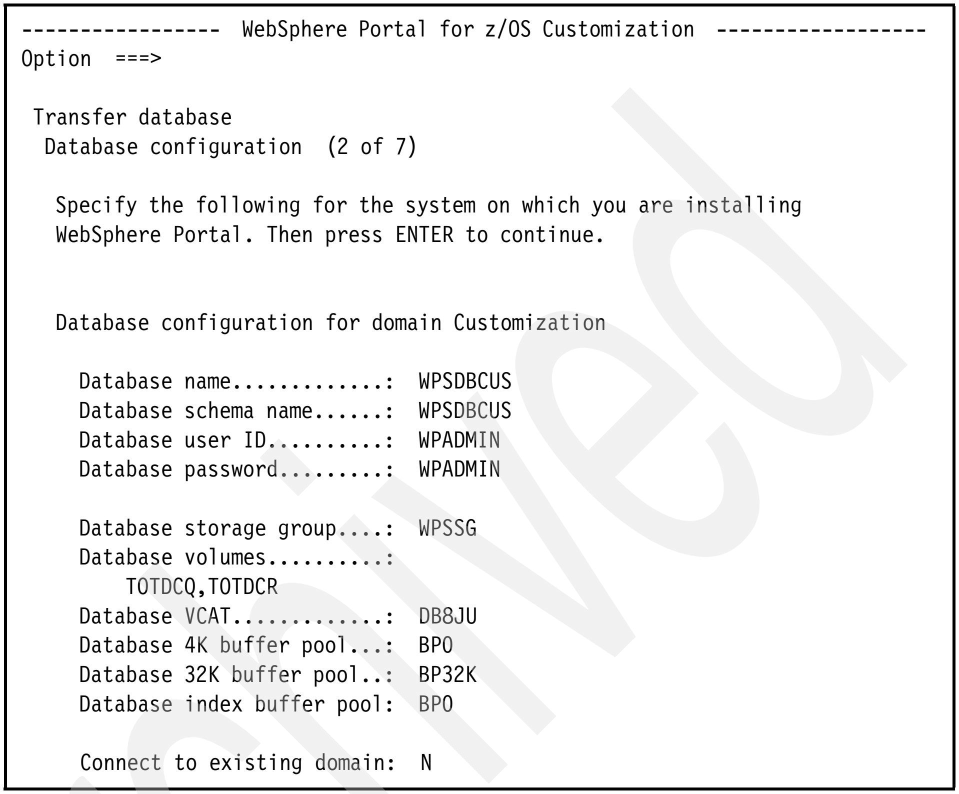
A total of 17 databases will be created.



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

 Transfer database
  Database configuration  (1 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain Release

     Database name............:  WPSDB
     Database schema name......:  WPSDB
     Database user ID.........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0
```

Figure 3-14   Customization panel - database configuration for domain Release

Panel 2 of 7 (shown in Figure 3-15) asks for all the same information as panel 1. This information is for the Customization database.



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

 Transfer database
  Database configuration  (2 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain Customization

     Database name.............:  WPSDBCUS
     Database schema name......:  WPSDBCUS
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0

     Connect to existing domain:  N
```

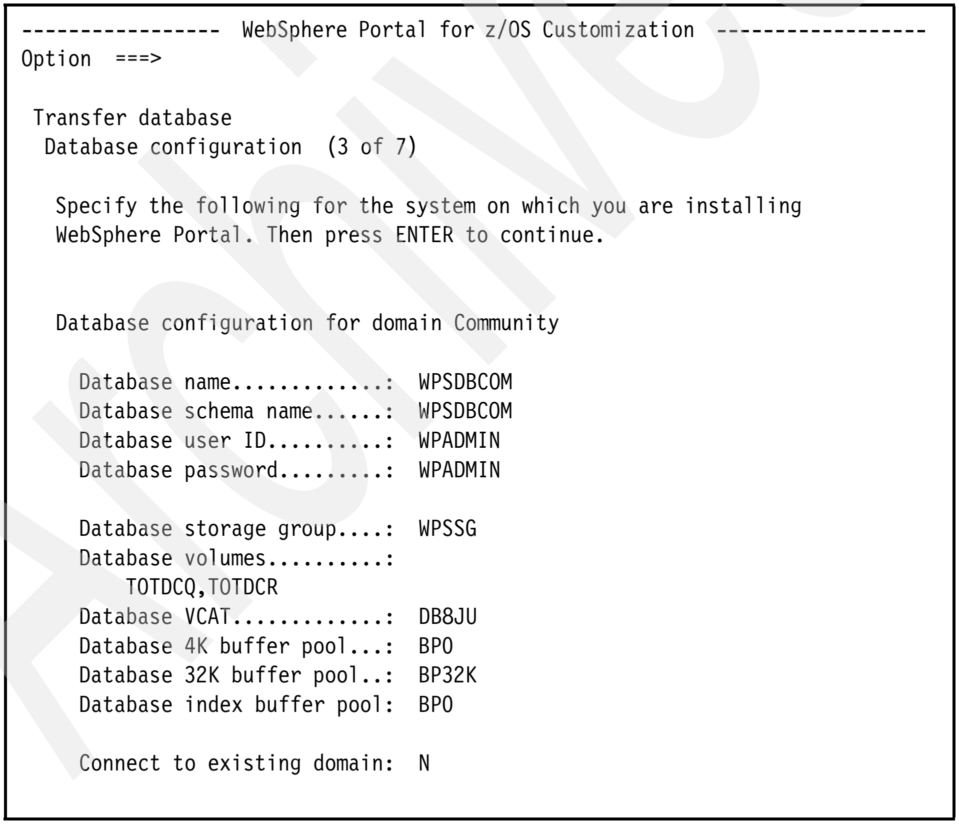
Figure 3-15 Customization panel - domain Customization

Panel 3 of 7 (shown in Figure 3-16) asks for much of the same information as panel 1, but there is one extra option to connect the database to an existing domain. This option is used only when you share database domains across different portal instances.

That is, if you have two completely independent WebSphere Application Server cells, and a portal serve or /cluster in each of them, these two portal servers/clusters can share some of their DB domains (for example, JCR). For more information about this topic, refer to “Sharing database domains between separate portal instances” at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/cu_dbdomnshare_zos.html

In our case, we chose `N` for `Connect to existing domain:` because this is the initial install. Panel 3 holds the information for the Community database.

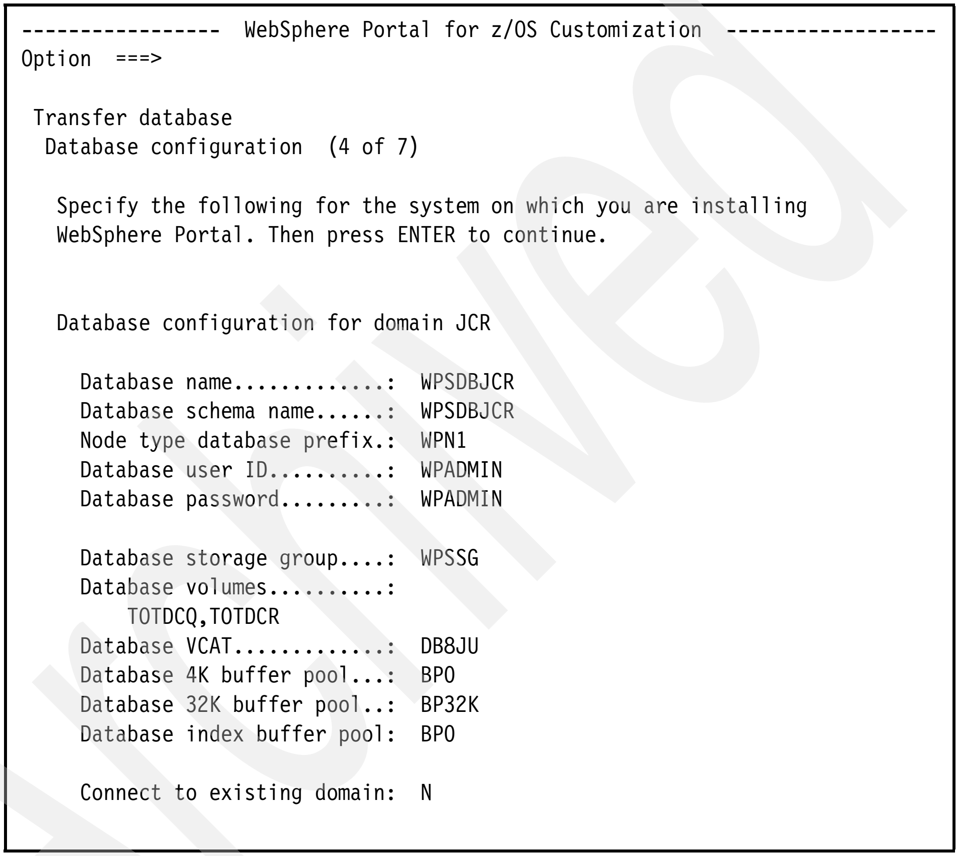


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Transfer database
  Database configuration  (3 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain Community

     Database name.............:  WPSDBCOM
     Database schema name......:  WPSDBCOM
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0

     Connect to existing domain:  N
```

Figure 3-16 Customization panel - domain Community

Panel 4 of 7 (shown in Figure 3-17) asks for much of the same information as panel 3, but this information is for the JCR database.

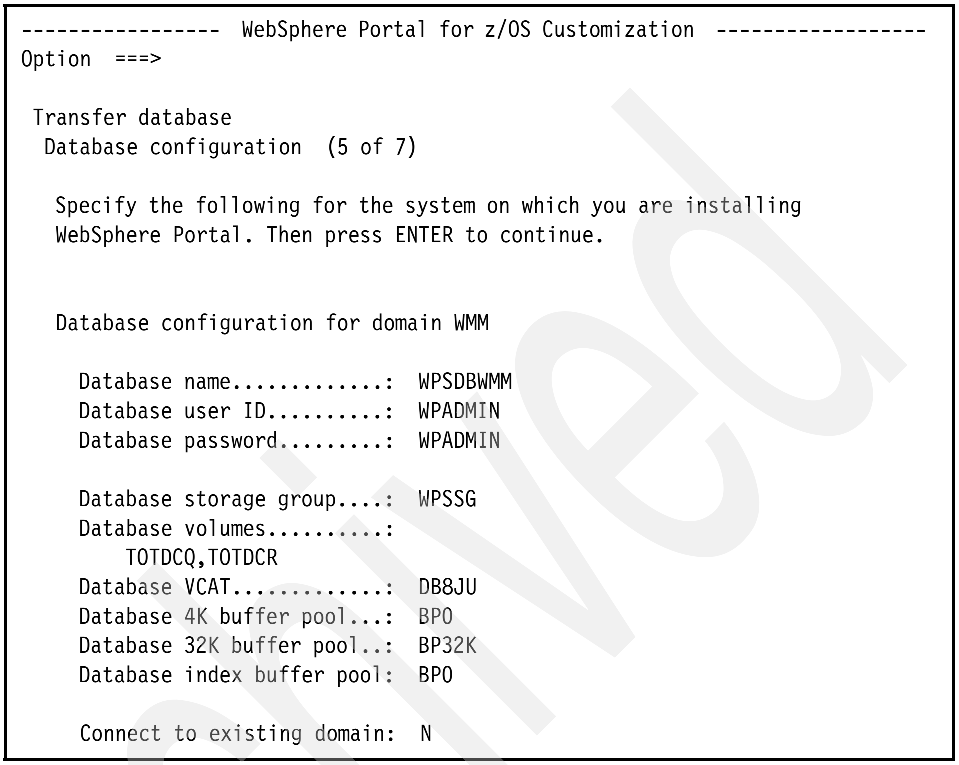
The one difference involves the `Node type database prefix`. One of the install jobs that gets created defines 10 databases with the 4-character prefix entered into this field. In our example, the database names that get created are WPN1DB01, WPNDB02, and so on.



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

 Transfer database
  Database configuration  (4 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain JCR

     Database name.............:  WPSDBJCR
     Database schema name......:  WPSDBJCR
     Node type database prefix.:  WPN1
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0

     Connect to existing domain:  N
```

Figure 3-17 Customization panel - domain JCR

Panel 5 of 7 (shown in Figure 3-18) asks for much of the same information as panel 3, but this information is for the WMM database.

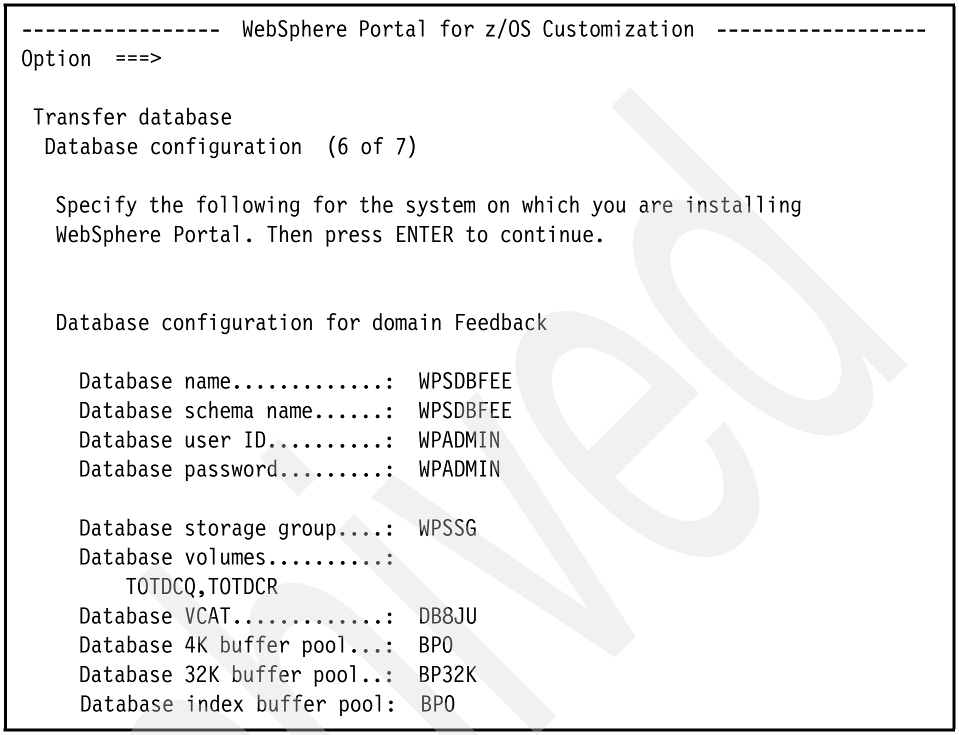


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Transfer database
  Database configuration  (5 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain WMM

     Database name.............:  WPSDBWMM
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0

     Connect to existing domain:  N
```

Figure 3-18   Customization panel - domain WMM

Note that the WMM database panel does not have a schema property to set, which means that the schema for the WMM database will be set to JAAS alias user ID for the WMM database.

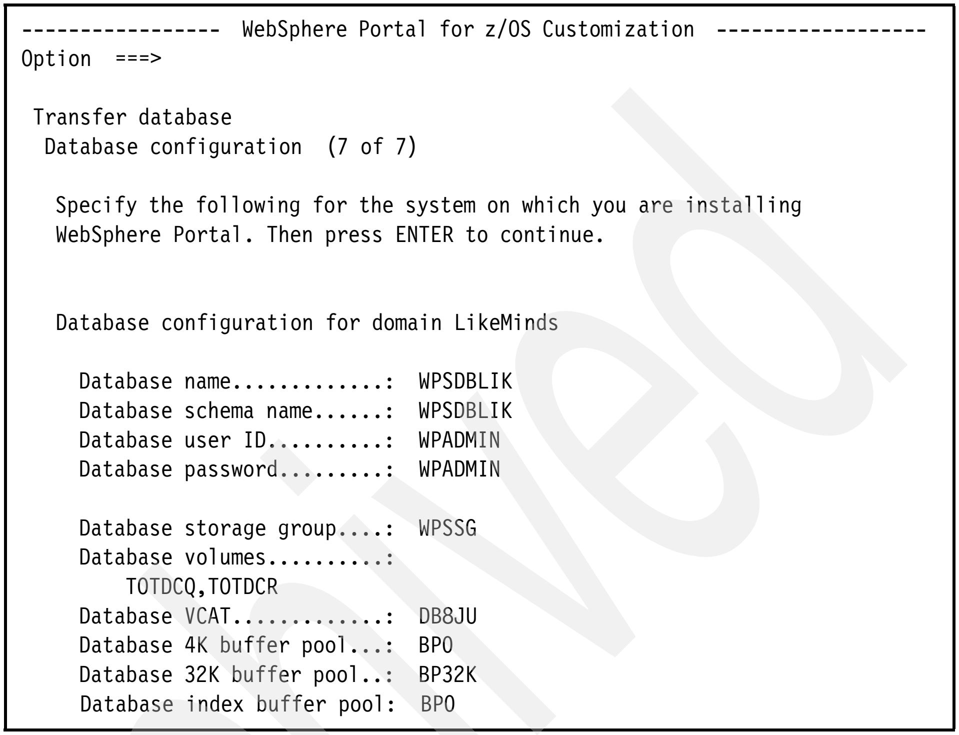
Panel 6 of 7 (shown in Figure 3-19) asks for all the same information as panel 1, but this information is for the Feedback database.



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

 Transfer database
  Database configuration  (6 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain Feedback

     Database name.............:  WPSDBFEE
     Database schema name......:  WPSDBFEE
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0
```

Figure 3-19 Customization panel - domain Feedback

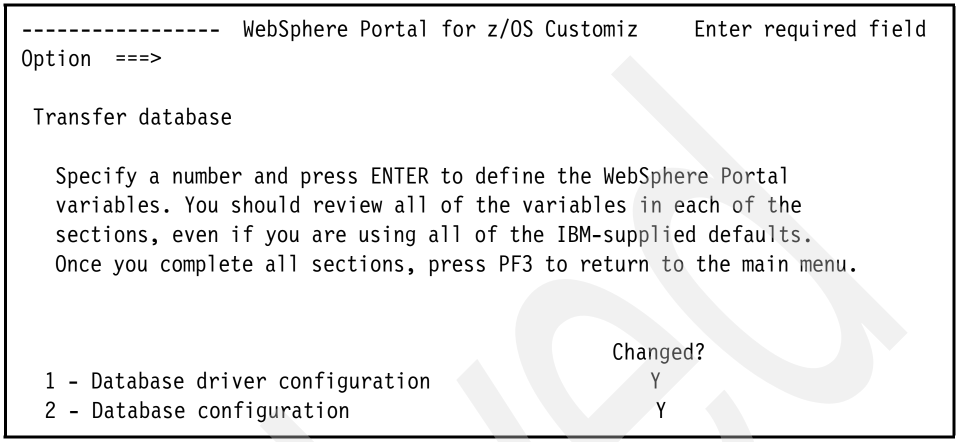
Panel 7 of 7 (shown in Figure 3-20) asks for all the same information as panel 1, but this information is for the LikeMinds database.



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

 Transfer database
  Database configuration  (7 of 7)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain LikeMinds

     Database name.............:  WPSDBLIK
     Database schema name......:  WPSDBLIK
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0
```

Figure 3-20 Customization panel - domain LikeMinds

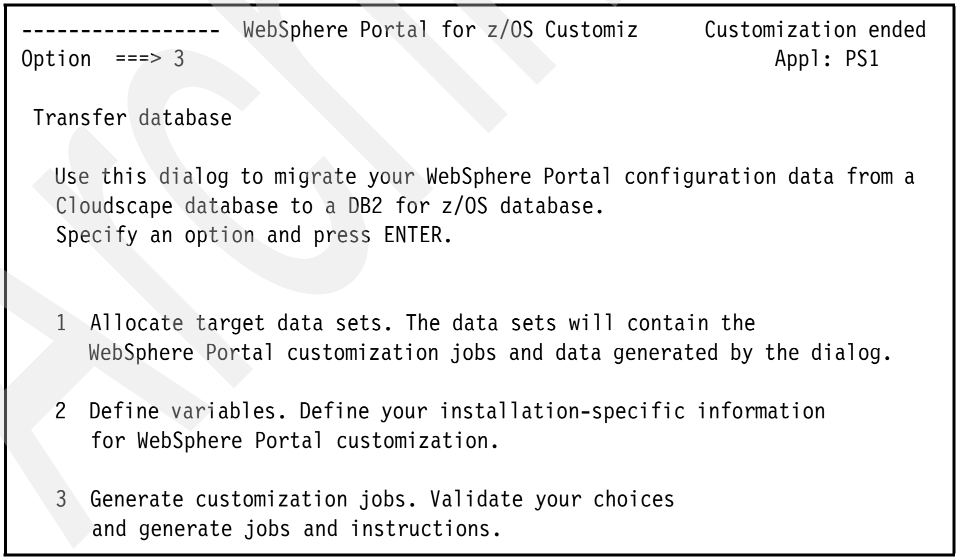
After all of the database configuration information has been entered into the panels, all the transfer database customization will show up as changed in the panel shown in Figure 3-21.



```
-----------------  WebSphere Portal for z/OS Customiz     Enter required field
Option  ===>

 Transfer database

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Database driver configuration                   Y
  2 - Database configuration                          Y
```

Figure 3-21 Customization panel - database transfer changed

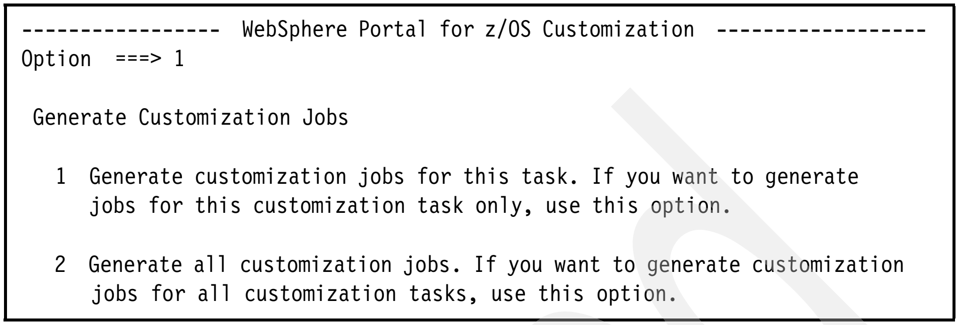
Select option **3** in the panel shown in Figure 3-22 to generate the batch jobs used to execute the database transfer steps.



```
-----------------  WebSphere Portal for z/OS Customiz      Customization ended
Option  ===> 3                                                   Appl: PS1

 Transfer database

   Use this dialog to migrate your WebSphere Portal configuration data from a
   Cloudscape database to a DB2 for z/OS database.
   Specify an option and press ENTER.


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

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

   3  Generate customization jobs. Validate your choices
      and generate jobs and instructions.
```

Figure 3-22 Customization panel - database transfer, generate customization jobs

Select option **1** in the panel shown in Figure 3-23 to generate only the jobs that relate to the database transfer task.

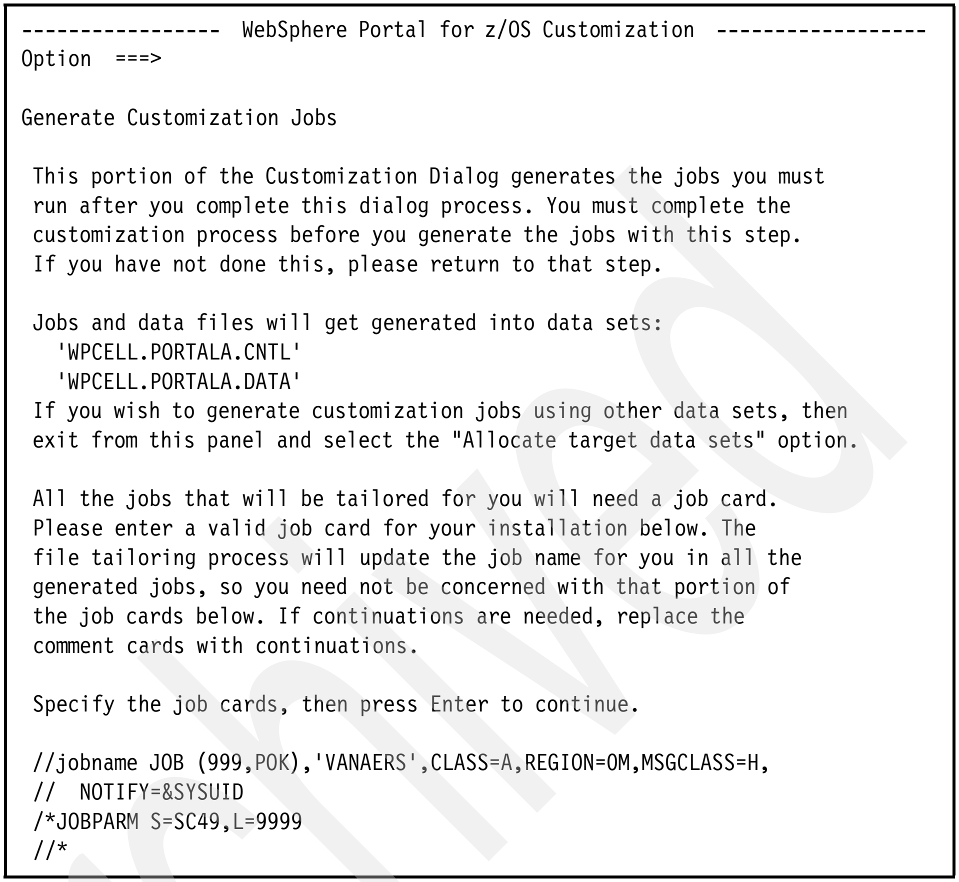


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1

 Generate Customization Jobs

   1  Generate customization jobs for this task. If you want to generate
      jobs for this customization task only, use this option.

   2  Generate all customization jobs. If you want to generate customization
      jobs for all customization tasks, use this option.
```

Figure 3-23   Customization panel - generate customization jobs for this task only

Make sure a valid job card is shown in the panel shown in Figure 3-24 on page 103.

> **Note:** Do *not* enter the USER and PASSWORD parameters with the DB2 user ID.
>
> The jobs that require the DB2 user and password will be generated with this information added to the job card.



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

Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'WPCELL.PORTALA.CNTL'
   'WPCELL.PORTALA.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.

 All the jobs that will be tailored for you will need a job card.
 Please enter a valid job card for your installation below. The
 file tailoring process will update the job name for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations.

 Specify the job cards, then press Enter to continue.

 //jobname JOB (999,POK),'VANAERS',CLASS=A,REGION=0M,MSGCLASS=H,
 //  NOTIFY=&SYSUID
 /*JOBPARM S=SC49,L=9999
 //*
```

Figure 3-24 Customization panel - generate customization jobs after dialog process is complete

After you press Enter, a display will appear showing all of the job names being generated; see Example 3-1. The jobs are copied into the HLQ.CNTL data set.

Example 3-1 Jobs successfully created

```
Processing for data set 'WPCELL.PORTALA.CNTL' ...

Member EJPIDBT successfully created.
Member EJPSRACD successfully created.
Member EJPSDRTB successfully created.
Member EJPSCRDB successfully created.
Member EJPSDBTV successfully created.
Member EJPSDBT successfully created.
Member EJPSDBTC successfully created.
```

```
Processing for data set 'WPCELL.PORTALA.DATA' ...

Member EJP2DBT successfully created.

***
```

3.2.3 Running the batch jobs

After all jobs have been customized with system-specific parameters, the jobs can be run. The jobs *must* be run in the order listed here.

1. EJPSRACD

   This job creates the WebSphere Portal database user and groups. The ID running this job must have the authority within RACF to create IDs and groups. A J2C authentication entry will be created in a later step which uses this ID to authenticate from WebSphere Application Server to the database back-end.

2. EJPSCRDB

   This job creates the database objects that are used by WebSphere Portal. Either run this job with an ID that has SYSADM, or set your SQLID to one that does at the beginning of the DDL statements.

   Because EJPSCRDB utilizes the DSNTIAD program, HLQ.RUNLIB.LOAD must be added to the JOBLIB. DSNTIAD is found in the DB2 runlib, and the job will fail if it cannot find it.

3. EJPSDBTV

   This job validates the connections to DB2. It tries to connect to each of the databases created in the EJPSCRDB step. Start the Deployment Manager and Node Agent before running this job.

   Note that EJPSDBTV can fail for two reasons:

   – The first reason is because either the universal UNIX Systems Services (USS) profile or the USS profile of the ID running the job does not have the proper DB2 paths defined.

     In this case, alter the universal profile (generally called /etc/profile) or the .profile of the ID running the job (in /u/userid). Add all the lines shown in Example 3-2 on page 105 with the system-specific paths. For PATH and LIBPATH, append the section in bold to the existing PATH and LIBPATH. Without these paths, EJPSDBTV will fail with a return code 256 in the first job step.

Example 3-2 Variables added to /etc/profile

```
PATH=/SC69/tools/ziptool:/usr/lpp/Printsrv/bin:/bin:/usr/lpp/db2/d8jg:.
LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib:/usr/lpp/db2/d8jg/jcc/lib/:
.
CLASSPATH=/usr/lpp/db2/d8jg/jcc/classes/sqlj.zip:$CLASSPATH
CLASSPATH=/usr/lpp/db2/d8jg/jcc/classes/db2jcc.jar:$CLASSPATH
CLASSPATH=/usr/lpp/db2/d8jg/jcc/classes/db2jcc_javax.jar:$CLASSPATH
CLASSPATH=/usr/lpp/db2/d8jg/jcc/classes/db2jcc_license_cisuz.jar:$CLASS
PATH
CLASSPATH=/etc/D8J1/DB2JccConfiguration.properties:$CLASSPATH
STEPLIB=DB8J8.SDSNLOAD:$STEPLIB
STEPLIB=DB8J8.SDSNLOD2:$STEPLIB
STEPLIB=DB8J8.SDSNEXIT:$STEPLIB
```

- The second reason why EJPSDBTV can fail is because WebSphere Portal's address spaces require DB2 SDSNLOAD and SDSNLOD2 to access DB2. These libraries must either be in the linklist, or they have to be added to the STEPLIB of the portal address spaces.

  If the address spaces cannot find the DB2 libraries, the following message will appear when trying to start the WebSphere Portal application server:

  ```
  EJPDB0013E: Internal Error.java.sql.SQLException: Failure in
  loading T2 native library
  ```

4. EJPSDBT

   After the databases are created and the connection to DB2 is valid, the data held in Cloudscape can be moved over to DB2. EJPSDBT recreates the tables defined in the Cloudscape databases in the DB2 databases defined by EJPSCRDB.

   After the tables are created, EJPSDBT copies the data from the Cloudscape tables into the DB2 tables. In addition, EJPSDBT creates a JDBC datasource in WebSphere Application Server. This job takes a while to run because it is moving a large amount of data from one DBMS to another.

   > **Note:** The data being transferred into DB2 is logged as it is inserted, so be prepared for a great deal of DB2 logging activity and make sure you have sufficient disk space to accommodate numerous archive logs.

5. After the Portal database has been successfully transferred, you must delete the XA transaction logs because they will contain records that still refer to the old Cloudscape database and you will suffer error messages related to transaction recovery if you try to start Portal while using the old tranlogs.

To delete the transaction logs, delete the directory <cell_name> under the <was_home>/tranlog directory but *not* the ../tranlog directory itself.

For example, on our system we deleted the wpcell directory (and everything under it) that was located here:

`/waswpconfig/wpcell/wpnode/AppServer/profiles/default/tranlog/`**`wpcell`**

6. EJPSDBTC

   This job runs the check data and runstats utilities against the WebSphere Portal tablespaces. The check data utility takes the tablespaces out of `copy pending` status. The runstats utility makes sure the tablespaces have the most current tablespace statistics.

   Make sure the ID running this job has the authority to run the check data and runstats utilities against the WebSphere Portal tables.

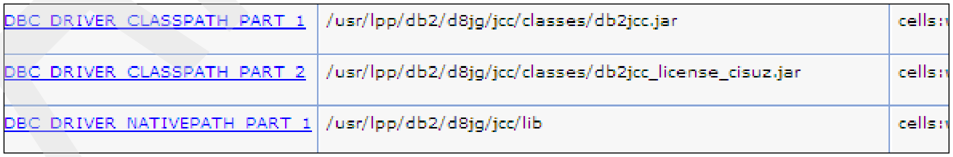
3.3 Post-database transfer information

WebSphere Portal can be started back up after all the jobs have been run and after you have deleted the old XA transaction log. After the server is up, it is good practice to make sure the WebSphere Portal logon is working. In this section, we describe a few items that you can check to verify that the Portal is utilizing DB2 instead of Cloudscape.

WebSphere Portal DB2 variables

The install jobs defined three variables for the classpaths of the DB2 libraries. Figure 3-25 shows these variables. Check to see if these exist by going to the WebSphere Application Server admin console and selecting **Environment** -> **WebSphere Variables**.

Note that these variables are defined on the node level, so make sure the scope is correct when you search for them.



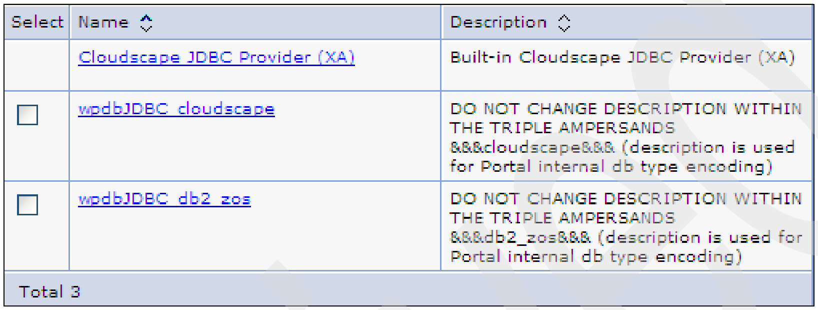
DBC_DRIVER_CLASSPATH_PART_1	/usr/lpp/db2/d8jg/jcc/classes/db2jcc.jar	cells:\
DBC_DRIVER_CLASSPATH_PART_2	/usr/lpp/db2/d8jg/jcc/classes/db2jcc_license_cisuz.jar	cells:\
DBC_DRIVER_NATIVEPATH_PART_1	/usr/lpp/db2/d8jg/jcc/lib	cells:\

Figure 3-25 WebSphere Portal DB2 variables

JDBC provider

The install jobs also created a JDBC provider; see Figure 3-26. Check to see if the wpdbJDBC_db2_zos JDBC provider exists by going into the WebSphere Application Server admin console and selecting **Resources** -> **JDBC Providers**.

This JDBC provider was defined at the cell level, so make sure the scope is correct when you search for it.



Select	Name	Description
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
□	wpdbJDBC_cloudscape	DO NOT CHANGE DESCRIPTION WITHIN THE TRIPLE AMPERSANDS &&&cloudscape&&& (description is used for Portal internal db type encoding)
□	wpdbJDBC_db2_zos	DO NOT CHANGE DESCRIPTION WITHIN THE TRIPLE AMPERSANDS &&&db2_zos&&& (description is used for Portal internal db type encoding)

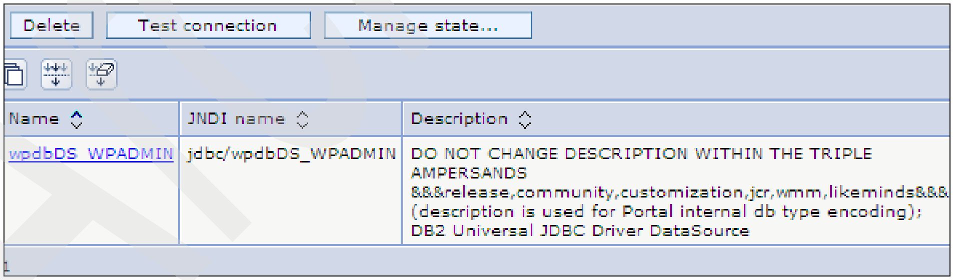
Total 3

Figure 3-26 WebSphere Portal JDBC Provider

DB2 datasource

Another item that can be verified is the DB2 datasource. Go into the WebSphere Application Server admin console and select **Resources** -> **JDBC Providers** -> **wpdbJDBC_db2_zos** > **data_sources**.

There should be one datasource defined here; see Figure 3-27.



Name	JNDI name	Description
wpdbDS_WPADMIN	jdbc/wpdbDS_WPADMIN	DO NOT CHANGE DESCRIPTION WITHIN THE TRIPLE AMPERSANDS &&&release,community,customization,jcr,wmm,likeminds&&& (description is used for Portal internal db type encoding); DB2 Universal JDBC Driver DataSource

1

Figure 3-27 DB2 datasource used by WebSphere Portal

For further information about the names and sizes of the tables created by the WebSphere Portal database transfer jobs, query the DB2 system catalog.

At this point, WebSphere Portal should be successfully configured to use DB2 as its back-end database repository.

4

Securing WebSphere Portal with LDAP

The initial install of WebSphere Portal is not a secure environment, and most users will want to enable more security in their Portal. Using an LDAP server for security allows centralized management of all users who can log on to the Portal.

This chapter describes how to set up WebSphere Portal to use an LDAP server for authentication, which is one of the possible choices. Another possibility would be a custom database registry.

This chapter discusses:

- Overview of basic WebSphere Portal security
- Using WebSphere Portal with WebSphere Application Server global security
- Installing and configuring the LDAP server on z/OS
- Enabling LDAP security

The topic of security is also addressed, in more detail, in Chapter 7, “Implementing integrated security” on page 225.

4.1 Portal security “out of the box”

You can configure Portal in a cell that already has global security enabled, but configuring Portal while security is enabled is another source of possible problems. Also, when building a Portal cluster, you must enable Portal security much earlier in the overall process if WebSphere global security is already enabled. Therefore, it is much easier to configure Portal without WebSphere global security being enabled and this is the approach we took.

Out of the box, whether WebSphere global security is enabled or not, Portal will use its *WebSphere Member Manager* (WMM) user registry to authenticate Portal users.

Before global security can be enabled in your WebSphere Application Server environment, you need to configure WebSphere Portal to function with a security registry. The most commonly used security registry is the Lightweight Directory Access Protocol (LDAP) server. We discuss basic security using LDAP in this chapter. LDAP allows authentication from multiple sources using the LDAP as a common repository for ID management.

WebSphere Portal can be set up using a database registry, an LDAP registry with realms, or an LDAP registry without realms.

In our scenario, the initial Portal configuration was performed with no security enabled. In this scenario, anyone can register as a new user and log in to the Portal. Some resources are secured, such as the Resource Permissions portlet, which is an administrator function. Configuring security is an advanced configuration task. You need to evaluate your security requirements during the planning phase to determine your requirements. For a more detailed discussion of this topic, refer to “Security” at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/fea_sec.html

After you decide to enable Portal security, there are several choices you need to evaluate. Portal security includes the following areas:

- Member services - discussed in 4.1.1, “Member services” on page 111
- Administration - discussed in 4.1.2, “Administration” on page 111
- Portal authentication - discussed in 4.1.3, “Portal authentication” on page 111
- Portal Authorization - discussed in 4.1.4, “Portal authorization” on page 112
- Delegated administration - discussed in 4.1.5, “Delegated administration” on page 114
- Java 2 security - discussed in 4.1.6, “Java 2 Security” on page 114

4.1.1 Member services

Many environments require centralized administration of user identities, credentials, and permissions. WebSphere Portal includes facilities for defining Portal users and managing user access rights. The user and group subsystem includes the following:

- Web pages where users can register and manage their own account information
- Administration portlets and XML Configuration Interface for managing user accounts and group information
- A repository that stores all the information about portal users

The default set of user profile attributes is based on the *inetOrgPerson* schema, which is supported by most Lightweight Directory Access Protocol (LDAP) directories.

The new concept of a realm allows you to aggregate users from one or more user registries and expose them as a coherent user population to WebSphere Portal; this is also referred to as *horizontal partitioning*.

4.1.2 Administration

Portal administrators or users (self-care) can create, delete, and modify users and groups. WebSphere Portal includes forms for registering new users and administration portlets for updating user and group information. You can also use the XML configuration interface to perform user management.

4.1.3 Portal authentication

Authentication is the process of establishing the identity of a user. Usually, the Portal uses the authentication services that are provided by IBM WebSphere Application Server. Another option is to use a third-party authentication server (such as IBM Tivoli Access Manager for e-business) that has a trusted association with the application server.

Authentication can be accomplished in several ways. It requires that users identify themselves to gain access to the Portal. Portal allows the following methods for login and authentication:

- Form-based authentication

  By default, Portal uses the Custom Form-based Authentication mechanism of WebSphere Application Server to challenge users to identify themselves. Users enter their user ID and password in the login portlet or the login screen.

- SSL client certificate authentication

  You can configure authentication using certificates stored in the user's browser or a SmartCard via Secure Sockets Layer (SSL) client certificate authentication. In this case, the authentication is done for users when they access the protected area of the Portal.

- Third party authentication

  You can use an external security manager such as IBM Tivoli Access Manager for e-business (TAM). When you configure external security, Portal trusts that the authentication was done by the third-party product.

- Automatic login with the login URL

  It is possible to log in to Portal using a static URL using a user ID and password. This method is suitable for automatic logon from a utility program to perform administrative functions.

See the InfoCenter for more detail on the various security configuration options at:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/cu_hpwkdf_zos.html`

4.1.4 Portal authorization

After determining the identity of the user, the Portal consults the *Portal Access Control* component to determine what resources (for example, pages and portlets) a user has permission to access. The Portal Access Control component stores the information in the WebSphere Portal database by default. Use the User Group Permissions and Resource Permissions portlets as well as the *XML Configuration Interface* and *Portal Scripting Interface* to manage the access control settings.

The Portal enforces access control to Portal assets, including portlets, pages, and user groups. You can also manage access control for specific resources in an external security manager, such as Tivoli Access Manager.

WebSphere Portal uses a role-based approach to manage user authorization for accessing portal resources. A *role* combines a set of permissions with a specific WebSphere Portal resource. This set of permissions is called a *role type*. By assigning the role to a user or a group, the user or users in the group are assigned permissions on the resource.

WebSphere Portal resources are part of a hierarchy (for example, a page can contain other pages or a folder that contains other folders or documents). Each resource in the hierarchy inherits the role assignments of its parent resource unless a *role block* exists. This inheritance reduces the administration overhead. When you assign a user (or a group the user is a member of) a role on a parent resource, the user automatically acquires that same role for all child resources.

Access control is based on role types and their corresponding allowed actions. The following role types are recognized:

- Administrator

  Unrestricted access to resources. This includes creating, configuring, and deleting resources. Administrators can also change the access control settings on resource.

- Security administrator

  Create and delete role assignments on resources.

- Delegator

  Assigning the delegator role to principals (users and groups) allows roles to be granted to them. Having the delegator role on other resources, such as specific portlets, is not useful.

- Manager

  Create new resources and configure and delete existing resources that are used by multiple users.

- Editor

  Create new resources and configure existing resources that are used by multiple users.

- Contributor

  View portal content and create new resources. The contributor role type does not include the permission to edit resources. It only allows a user to create new resources.

- Privileged user

  View portal content, customize portlets and pages, and create new private pages.

- User

  Viewing portal content (for example, viewing a specific page).

- No role assigned

  Cannot interact with resource.

4.1.5 Delegated administration

WebSphere Portal supports delegated access control administration. An *administrator* is a user who is authorized to modify the access control configuration by changing role assignments and creating or deleting role blocks. Administrators can delegate specific subsets of their administrative privileges to other users or groups. These users or groups can in turn delegate subsets of their privileges to additional users and groups.

4.1.6 Java 2 Security

Java 2 (J2SE™) security provides a policy-based, fine-grain access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. J2SE security allows you to set up individual policy files that control the privileges assigned to individual code sources.

4.2 Using WebSphere Portal with WebSphere Application Server global security

In this scenario we used an LDAP server running on z/OS. Whether we used LDAP on z/OS or on a distributed platform would not make any difference, from an application point of view. LDAP is always approached in a client/server mode.

However, it makes a difference in quality, availability, and performance. The concepts used for this LDAP server are very similar to many other LDAP servers that are available. WebSphere Portal uses the security source defined in the WebSphere Application Server. It is not possible to enable security on Portal without having it enabled in WebSphere Application Server first.

After WebSphere Portal and WebSphere Application Server have been configured to use LDAP, you will log on to your Portal using the LDAP ID instead of the default wpsadmin ID. Authorities within Portal will be granted to IDs that are defined on the LDAP server. There will no longer be the option to create IDs via the WebSphere Portal administrative Web pages unless the Portal administrative ID is also an LDAP administrator.

4.3 Installing the LDAP server

In the following sections we explain how an LDAP server can be set up, if you do not already have one in your environment.

Prior to z/OS 1.8, the IBM Secureway LDAP Server was provided with z/OS. However, in z/OS 1.8 you can also choose to configure Tivoli Directory Server for z/OS.

In this section we describe how to configure the IBM Secureway LDAP Server, but the procedure for configuring a Tivoli Directory Server for z/OS is very similar. For more information about that topic, refer to *Tivoli Directory Server for z/OS V1R8.0 Administration and Use,* SC23-5191, which can be downloaded from the following site:

http://www.ibm.com/servers/eserver/zseries/zos/bkserv/r8pdf/itds.html

4.3.1 Customizing the LDAP profiles

Before you can start securing the Portal, you must configure an LDAP server for your environment. The first step is to copy four LDAP profile files and two schema files from the LDAP product directory to a working directory where you will edit them. The four LDAP profile files contain variable/value pairs that you must review and customize in order to set a range of properties that determine the names used by the LDAP Server and how it interfaces with the network, RACF, and DB2.

After editing the four LDAP profile files you will run the ldapcnf utility, which takes the LDAP profile files as input and then generates JCL, DB2 DDL statements and other tailored property files that will be used by the LDAP Server. The two schema files, schema.user.ldif and schema.IBM.ldif, contain the definitions of the various attributes and their relationships.

After editing the schema files, you will load the schemas into LDAP. Copy the following files from /usr/lpp/ldap/etc/ to a work directory (such as a user home) where they can be saved. Copy the following files:

```
ldap.db2.profile
ldap.profile
ldap.racf.profile
ldap.slapd.profile
schema.user.ldif
schema.IBM.ldif
```

Edit the files to include system-specific values.

Start by editing the ldap.db2.profile. LDAP exports several registry views, one of which is the TDBM view. The TDBM view represents a real hierarchical directory structure, but currently completely materialized by DB2 tables.

The installation scripts must know about the DB2 installed on the system where LDAP information will be stored. The most important values are listed here, but others may have to be changed based on system specifics.

- DB2_LOCATION is the DDF location name of the DB2 subsystem that will be used to store LDAP data.
- TDBM_DB2_USERID is the ID that will own the DB2 tables created for LDAP. This ID should also be the ID that the LDAP started task will run with.
- TDBM_DB2_DBNAME is the name that is given to the database used by LDAP when it is created in DB2.
- TDBM_DB2_STORAGEGROUP is the storage group that used for the LDAP tables in DB2.

Example 4-1 shows our version of the ldap.db2.profile file.

Example 4-1 ldap.db2.profile

```
# Description:
#    DB2 server location (or data source).
#---------------------------------------------------------------------
DB2_LOCATION='DB8J'
#---------------------------------------------------------------------
# Description:
#    MVS database owner ID. This ID will be the
#    high level qualifier for the DB2 backend tables.
#---------------------------------------------------------------------
TDBM_DB2_USERID='LDAPSRV'
#---------------------------------------------------------------------
# Description:
#    DB2 tables for the TDBM backend.
#---------------------------------------------------------------------
TDBM_DB2_DBNAME='PSLDAPDB'
#---------------------------------------------------------------------
# Description:
#    Storage group name that will contain the
#    TDBM backend tablespaces.
#---------------------------------------------------------------------
TDBM_DB2_STORAGEGROUP='PSLDAPSG'
```

Next, update the ldap.racf.profile. The only information required is the UNIX System Services (USS) UID and GID of the ID being used to run the LDAP started task.

Note: We do not recommend running the LDAP server with a user and group that has a UID and GID of zero (0). They are only specified as 0 here since it is a test environment.

Example 4-2 shows our version of the ldap.racf.profile file.

Example 4-2   ldap.racf.profile

```
# Description:
#    This option establishes the group ID of the user ID
#    that the LDAP started task will run under.
#----------------------------------------------------------------------
LDAPGID='0'
#----------------------------------------------------------------------
# Description:
#    This option establishes the user ID number of the user ID that
#    the LDAP started task will run under.
#----------------------------------------------------------------------
LDAPUID='0'
```

Now update the ldap.slapd.profile. Note the following points:

- LDAP_HOSTNAME is the domain name of the LPAR that the LDAP server is running on.
- PORT is the non-SSL port that the LDAP server will listen on for connection requests.
- Enter a SECUREPORT if the system is utilizing SSL.

There are several other parameters but the ones updated in our scenario are the only ones described.

Example 4-3 shows our version of the ldap.slapd.profile file.

Example 4-3   ldap.slapd.profile

```
# Description:
#   This setting indicates the host name or IP address that
#   the LDAP server will listen for incoming client requests.

# Option Updated in SLAPDCNF:
#   listen
#----------------------------------------------------------------------
LDAP_HOSTNAME='wtsc49.itso.ibm.com'
#----------------------------------------------------------------------
# Description:
```

```
#   The setting indicates what TCP/IP socket port
#  the server will listen on for incoming non-SSL connections.
# #
# Option Updated in SLAPDCNF:
#   listen
#-----------------------------------------------------------------------
PORT='29389'

#-----------------------------------------------------------------------
# Description:
#   The setting indicates what TCP/IP socket port
#   the server will listen on for incoming SSL connections.
#
# Option Updated in SLAPDCNF:
#   listen
#-----------------------------------------------------------------------
#SECUREPORT='29689'
```

The last .profile file that has to be customized is the ldap.profile. This is the largest file with the most parameters to update. Here is a list of the parameters that were altered during our install.

- OUTPUT_DATASET is a PDS that gets created to store the output from the LDAP configuration utility.
- GLDHLQ is the high level qualifier of the LDAP server data sets.

The next parameters are the high level qualifier (HLQ) and volume where the data set resides for three DB2 data sets (SDSNEXIT, SDSNLOAD and DBRMLIB).

- DSN_SSID is the subsystem ID for the DB2 being used for the LDAP database.
- CSSLIBVOL and LINKLIBVOL refer to the volume where the system SYS1.CSSLIB and SYS1.LINKLIB reside.
- LDAPUSRID is the RACF ID used to run the LDAP address space.
- LDAPUSRGRP is the RACF group that the LDAPUSRID is a member of.
- TDBM_SUFFIX is the suffix used as the origin of the LDAP hierarchy.

  Figure 4-2 shows the full LDAP hierarchy defined with `o=sc49portal` as the top of hierarchy.

- ADMINDN is the common name of the ID that is entered into LDAP as the LDAP administrator.
- All of the job card variables will be used in the JCL for running the LDAP install jobs.

- ${SOURCE_CMD} lines should contain the path where the .profile files are stored.

Example 4-4 shows our version of the ldap.profile file.

Example 4-4 ldap.profile

```
OUTPUT_DATASET='WPCELL.LDAP.CNFOUT'
GLDHLQ='SYS1'
DSN_SDSNEXITHLQ='DB8JU'
SDSNEXITVOL=’TOTDDT’
DSN_SDSNLOADHLQ='DB8J8'
SDSNLOADVOL=’TOTDDT’
DSN_SDSNDBRMHLQ='DB8JU'
DSN_SSID='D8J1'
CSSLIBVOL='Z17RB1'
LINKLIBVOL='Z17RB1'
LDAPUSRID='LDAPSRV'
LDAPUSRGRP='LDAPGRP'
TDBM_SUFFIX='o=sc49portal'
ADMINDN='cn=LDAPADM'
APF_JOBCARD_1="//LDAPCFGA  JOB NOTIFY=&SYSUID,CLASS=A,"
APF_JOBCARD_2="//   MSGCLASS=H,REGION=0M,TIME=1440"
PRGCTRL_JOBCARD_1="//LDAPCFGP  JOB NOTIFY=&SYSUID,CLASS=A,"
PRGCTRL_JOBCARD_2="//   MSGCLASS=H,REGION=0M,TIME=1440"
DB2_JOBCARD_1="//LDAPCFGD  JOB NOTIFY=&SYSUID,CLASS=A,"
DB2_JOBCARD_2="//   MSGCLASS=H,REGION=0M,TIME=1440"
RACF_JOBCARD_1="//LDAPCFGR  JOB NOTIFY=&SYSUID,CLASS=A,"
RACF_JOBCARD_2="//   MSGCLASS=H,REGION=0M,TIME=1440"
${SOURCE_CMD} ${USR_LPP_ROOT}/u/houde/ldap.slapd.profile
${SOURCE_CMD} ${USR_LPP_ROOT}/u/houde/ldap.db2.profile
${SOURCE_CMD} ${USR_LPP_ROOT}/u/houde/ldap.racf.profile
```

When all the LDAP variables have been customized for its environment, you can fun the ldapcnf utility. Change the directory to /usr/lpp/ldap/sbin/. Run the utility from this directory using the following syntax.

```
./ldapcnf -i /[custom]/[path]/ldap.profile
```

After you press Enter, the output shown in Example 4-5 will display.

Example 4-5 ldapcnf script output

```
The utility is finished checking for errors.
Generating tdbSpufi ....
Finished generating tdbSpufi.
```

```
Generating dbCli ....
Finished generating dbCli.
Generating dsnaoini ....
Finished generating dsnaoini.
Generating ldapSrvProc ....
Finished generating ldapSrvProc.
Generating slapdcnf ....
Finished generating slapdcnf.
Generating irr ....
Finished generating irr.
Generating kerb ....
Finished generating kerb.
Generating slapdenv ....
Finished generating slapdenv.
Generating racf ....
Finished generating racf.
Generating prgmCtrl ....
Finished generating prgmCtrl.
Generating ocsfApf ....
Finished generating ocsfApf.
Generating ocsf ....
Finished generating ocsf.
Generating gldOcsfApf ....
Finished generating gldOcsfApf.
Generating PROGxx ....
Finished generating PROGxx.
Generating apf ....
Finished generating apf.
Exiting with return code 0.
```

The customized output has to be picked up from the members in HLQ.CNFOUT.

4.3.2 Post-customization tasks

Perform the following tasks after customization:

- Copy the started task procedure that was generated by the ldapcnf utility into the system procedure library. In our scenario, we wanted our LDAP server to be called LDAPWPS so we copied member LDPSRV to SYS1.PROCLIB as member LDAPWPS. This procedure LDAPSRV” was created as a member in HLQ.CFOUT.
- Submit the job HLQ.CNFOUT(RACF). This job issues RACF commands to grant all the authority that is required by the LDAP administrator ID within RACF.

- Job DBCLI binds plan DSNACLI. This plan may already be bound on the system, because it is a common DB2 plan. Check to see if it is already bound by running the following select:

  ```
  SELECT * FROM SYSIBM.SYSPLAN WHERE NAME LIKE 'DSNACLI%’;
  ```

  If the DB2 subsystem being used is not defined with MIXED=YES in ZPARM for valid mixed and graphic ascii ccsids, package DSNCLIMS will fail with the error shown in Example 4-6.

Example 4-6 Text from DB2 data set HLQ.SDSNSAMP(DSNTIJCL)

```
AFTER THE APPLICATION OF PTF UQ87850, THE BINDING OF DSNCLIMS
TO A MIXED=NO SUBSYSTEM WILL RESULT IN SQLCODE=-189:
   DSNX200I  = BIND SQL ERROR
              USING SYSADM AUTHORITY
              PLAN=(NOT APPLICABLE)
              DBRM=DSNCLIMS
              STATEMENT=747
              SQLCODE=-189
              SQLSTATE=22522
              TOKENS=65534
              CSECT NAME=DSNXOOS2
              RDS CODE=-840
BIND WITH SQLERROR(CONTINUE) TO BYPASS THIS ERROR.  HOWEVER
IGNORING THIS ERROR MEANS YOU WILL NOT BE ABLE TO FETCH FROM AN
ASCII DBCLOB COLUMN USING THE SQLGETDATA() API OR THROUGH LOB
LOCATORS UNTIL YOU DEFINE YOUR SUBSYSTEM MIXED=YES WITH VALID
MIXED AND GRAPHIC ASCII CCSIDS, FOLLOWED BY A REBIND OF DSNCLIMS.
```

  Fix or bypass this error by adding SQLERROR(CONTINUE) to the BIND PACKAGE statement:

  ```
  BIND PACKAGE (DSNACLI) ACTION(REPLACE) -

    MEMBER(DSNCLIMS) KEEPDYNAMIC(YES) SQLERROR(CONTINUE)
  ```

  The ID running this job will need to have the authorization to bind plans and packages in the DB2 subsystem.

- HLQ.CNFOUT member TDBSPUFI should be run in SPUFI to create all of the DB2 objects used by the LDAP server. The storage group definition was not generated when we ran the ldapcnf script, so it had to be added manually. The DDL will not run successfully without the storage group definition since the CREATE DATABASE DDL specifies a storage group.

  ```
  CREATE STOGROUP stogroupname VOLUMES(vol1,vol2) VCAT your_vcat;
  COMMIT;
  ```

4.3.3 LDAP server address space

Now start the LDAP server, as follows:

```
/START LDAP_PROC_NAME
```

Examine the messages for the server that is starting up and verify that it started correctly. In our case, as shown in Example 4-7, the messages indicated that the LDAP server was started in single mode.

Example 4-7 Messages for server

```
IEF695I START LDAPWPS  WITH JOBNAME LDAPWPS  IS ASSIGNED TO USER LDAPSRV , GROUP
LDAPGRP

$HASP373 LDAPWPS  STARTED

IEF403I LDAPWPS - STARTED - TIME=12.04.39 - ASID=00D1 - SC49

GLD0022I z/OS Version 1 Release 6 Integrated Security Services LDAP  098

Server

Starting slapd.

GLD0122I Slapd is ready for requests.  099
```

At this point, the LDAP structure is not populated yet and the schema structure of the LDAP hierarchy is not entered. The current administration user of LDAP is a temporary admin, which we will change. The following two files are delivered with LDAP on z/OS. We copied them, but they have to be adapted:

- Edit /......../schema.user.ldif
- Change <suffix> to your dn: cn=schema, **o=sc49portal**
- Edit /......../schema.IBM.ldif
- Change <suffix> to your dn: cn=schema, **o=sc49portal**
- Run the command from open MVS, as follows:

  ```
  ldapmodify -h wtsc49.itso.ibm.com -p 29389 -D 'cn=LDAPADM' -w secret
  -f schema.user.ldif
  ```

  The values you code for the `-D` and `-w` properties must match the values you coded for the AdminDN and AdminPW properties in HLQ.CNFOUT(SLDAPCNF).

  Change to the directory where you have your customized schema files before entering the following ldapmodify commands.

- Run command from OMVS, as follows:

  ```
  ldapmodify -h wtsc49.itso.ibm.com -p 29389 -D 'cn=LDAPADM' -w secret
  -f schema.IBM.ldif
  ```

- Search ldap by issuing (from OMVS):

  ```
  ldapsearch -h wtsc49.itso.ibm.com -p 29389 -V 3 -s base -b ""
  "objectclass
  ```

- Prepare the LDIF file to populate the LDAP schema. If you cannot reuse LDIF files from an existing installation to use as an example, you will have to build them. Example 4-8 shows an excerpt of the LDIF file we used for loading the data.

> **Note:** Be aware that Example 4-8 does not show the complete ldif.
>
> The entries for the RACF user IDs and groups of the cell's started task user IDs, the WebSphere Application Server Admin user ID, and the unauthenticated user ID and its group required for enabling WebSphere Application Server global security are not shown here.
>
> Without those entries, WebSphere Application Server will not initialize when security has been enabled.

Example 4-8 Sample lines from portal.ldif

```
dn: o=sc49portal
o: WPS_ROOT
objectclass: top
objectclass: domain
dc: o=sc49portal
description: WPS USER REGISTRY
userPassword: wpsbind
ownersource: o=sc49portal
entryowner: access-id:uid=wpsbind,cn=users,o=sc49portal
aclpropagate: TRUE
ownerpropagate: TRUE
aclsource: o=sc49portal
aclentry:
access-id:uid=wpsbind,cn=users,o=sc49portal:normal:rwsc:object:ad
aclentry: group:CN=ANYBODY:normal:rsc

dn: uid=LDAPADM,o=sc49portal
```

```
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: ibm-nativeAuthentication
uid: LDAPADM
userpassword: secret
ibm-nativeId: LDAPADM
sn: LDAPADM
givenName: LDAPADM
cn: LDAPADM

dn: cn=users,o=sc49portal
objectclass: container
objectclass: top
cn: users

dn: cn=groups,o=sc49portal
objectclass: top
objectclass: container
cn: groups

dn: uid=wpsadmin,cn=users,o=sc49portal
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: wpsadmin
userpassword: wpsadmin
sn: admin
givenName: wps
cn: wps admin

dn: cn=wpsadmins,cn=groups,o=sc49portal
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=wpsadmin,cn=users,o=sc49portal
cn: wpsadmins
```

- Run the following command from OMVS:

```
ldapadd -h wtsc49.itso.ibm.com -p 29389 -D 'cn=LDAPADM' -w secret
-f /waswpconfig/wpcell/wpport/Portal/ldapwps/portal.ldif
```

You should see output like this:

```
adding new entry uid=wpsadmin,cn=users,o=sc49portal
......
```

This populated the LDAP with users and groups.

- Add Web Content Management (WCM) users and groups if required in the the same way.
- Edit WPCELL.LDAP.CNFOUT(SLAPDCNF)

  Now that you have added the users to the LDAP database, you can use them to log in to LDAP. Also, the LDAP administrator user ID, which must be specified on properties AdminDN and AdminPW in the SLAPDCNF in order to initialize the LDAP server, can now be changed to a user ID within the LDAP database.

  This is so that the LDAP Administrator's password is not held in the SLAPDCNF where it might be read by someone. By defining the LDAP Administrator in the LDAP database you can then encrypt the password, or define the LDAP administrator so its password is held in RACF.
- Make this change in the SLAPCNF member:

  ```
  Old - #adminDN "cn=LDAPADM"
  new - adminDN "uid=LDAPADM,o=sc49portal"
  ```
- Recycle the LDAP server.

  From here the new adminDN is used with its password.
- Browse the LDAP content with any of the available LDAP browsers.

  You can find a great deal of freeware able to browse the contents of the LDAP database. Remember that this browser is always a TCP/IP client for the server.

  Here are a few examples of freeware LDAP browsers:

  – LDAPBrowser
  – SoftTerra

Figure 4-1 on page 126 shows the hierarchical organization of the data in the LDAP database. This figure can be directly related to the portal.ldif in Example 4-8 on page 123; portal.ldif shows how we defined each of the boxes in Figure 4-1 on page 126.

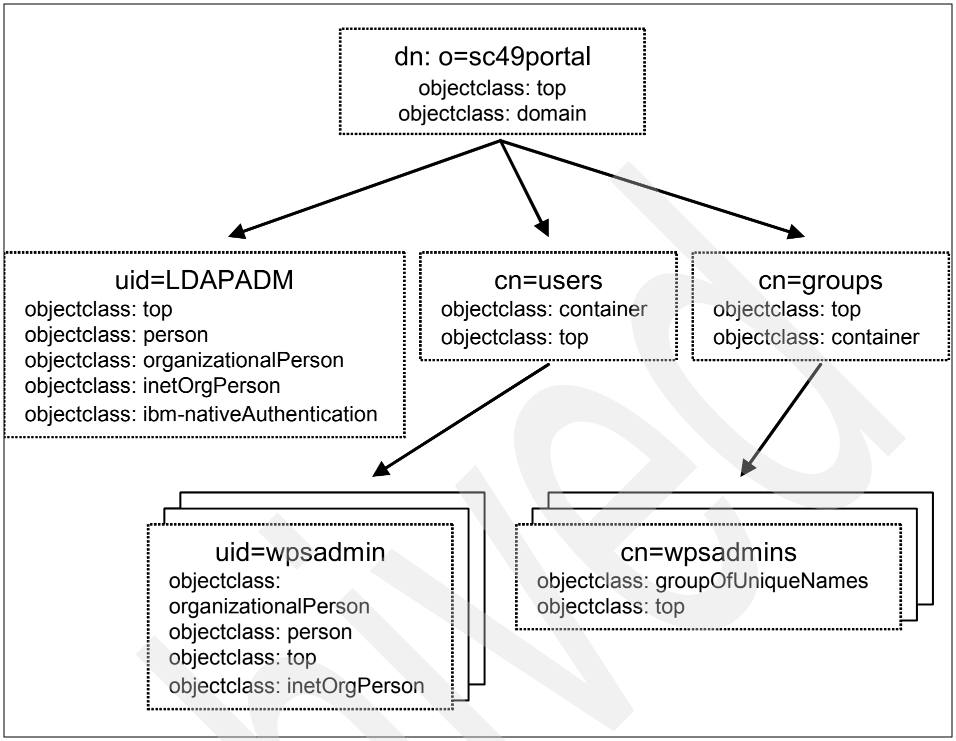


Figure 4-1 LDAP architecture

The distinguished name of the hierarchy starts with `o=sc49portal`. Under o=sc49portal is a single LDAP administrator ID called LDAPADM. There can only be one LDAP administrator ID.

At the same level there are two containers: *users* and *groups*.

- The users container can point to multiple users.
- The groups container points to multiple groups. In our case, we only defined one group called wpsadmins. This group is the Portal administrators group. The wpsadmin user was added to the wpsadmins group in the portal.ldif using the uniquemember tag:

  ```
  uniquemember: uid=wpsadmin,cn=users,o=sc49portal
  ```

At this point, the LDAP server is now fully configured. The last thing to do is verify that the LDAP server is functioning correctly by searching the ldap database for the IDs and groups that were entered into it.

Search the LDAP database by using the **ldapsearch** command:

```
ldapsearch -h ldap_hostname -p ldap_server_port -s base
objectclass=*

ldapsearch -h wtsc49.itso.ibm.com -p 29389 -s base objectclass=*
```

Tip: It is helpful to use an LDAP browser to verify the contents of the LDAP.

4.4 Enabling LDAP security in Portal

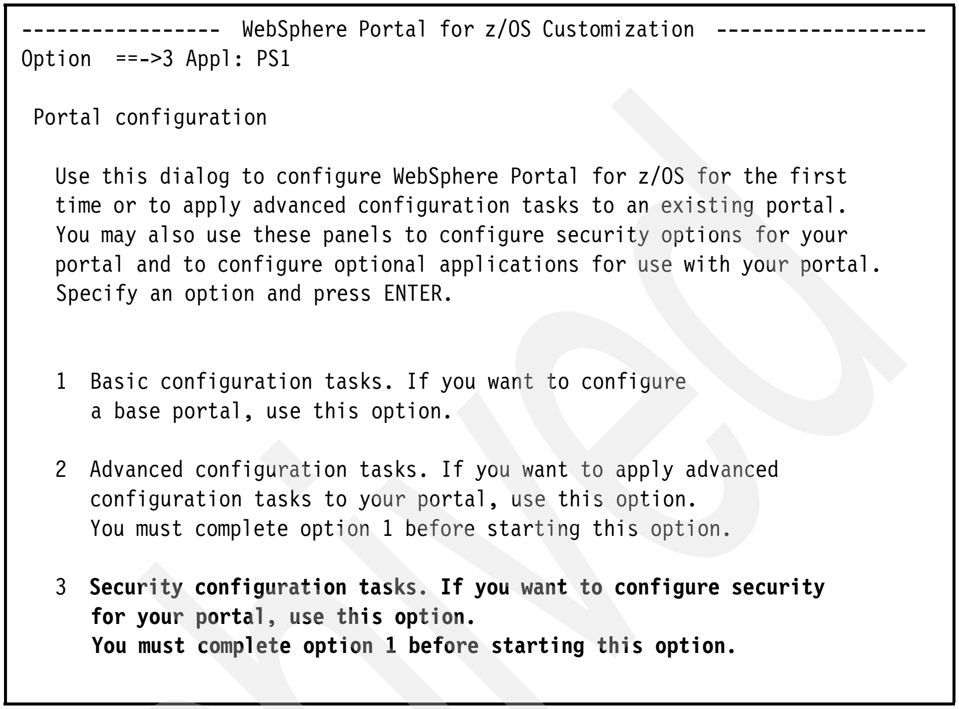
With an LDAP server in place, WebSphere Portal and WebSphere Application Server can now be secured. WebSphere Portal cannot be secured without securing WebSphere Application Server as well. Many of the values that are entered into the customization dialog are intended for securing WebSphere Application Server.

Start the WebSphere Application Server install dialog in the same way as done in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35 and Chapter 3, “Migrating the Portal database from Cloudscape to DB2” on page 83. Select option **5** → **1**, as shown in Figure 4-2.

```
5  WebSphere Application Server-based add-on products. Configure
   other products that are built on WebSphere Application Server.
-----
1   WebSphere Portal for z/OS
    Configure WebSphere Portal
-----
```

Figure 4-2 Selecting WebSphere Portal configuration from the WebSphere Application Server Customization panel

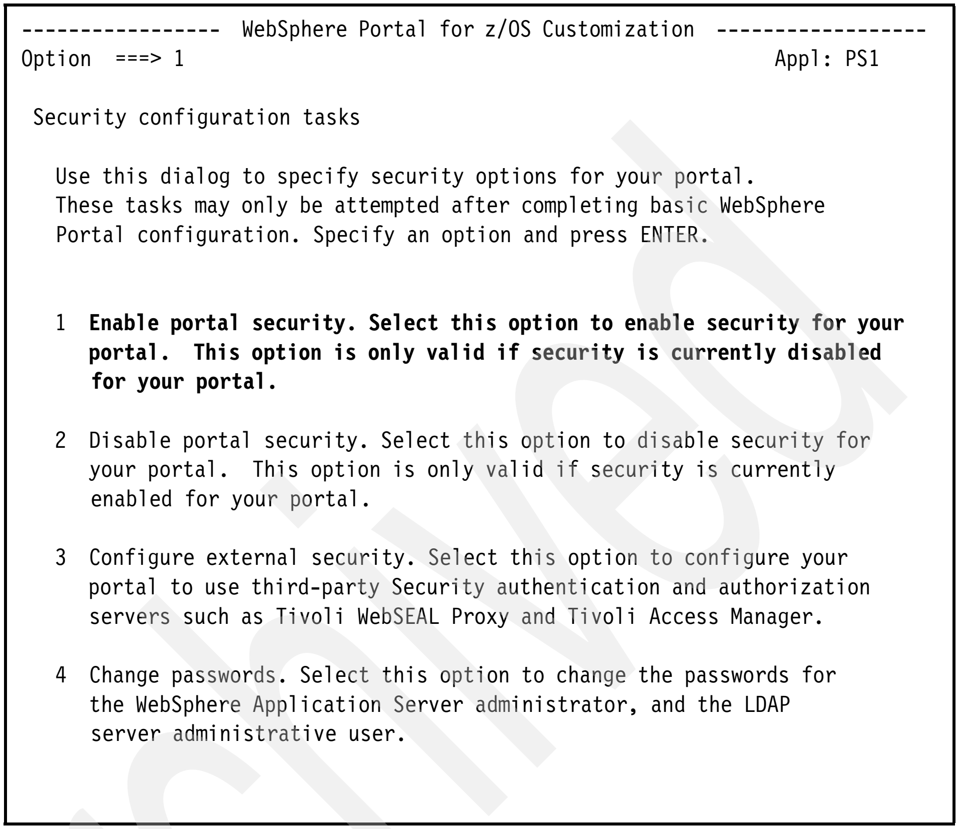
In the next panel, shown in Figure 4-3, choose option **3 Security configuration tasks**.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ==->3 Appl: PS1

 Portal configuration

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time or to apply advanced configuration tasks to an existing portal.
   You may also use these panels to configure security options for your
   portal and to configure optional applications for use with your portal.
   Specify an option and press ENTER.


   1  Basic configuration tasks. If you want to configure
      a base portal, use this option.

   2  Advanced configuration tasks. If you want to apply advanced
      configuration tasks to your portal, use this option.
      You must complete option 1 before starting this option.

   3  Security configuration tasks. If you want to configure security
      for your portal, use this option.
      You must complete option 1 before starting this option.
```

Figure 4-3 Customization panel - security configuration tasks

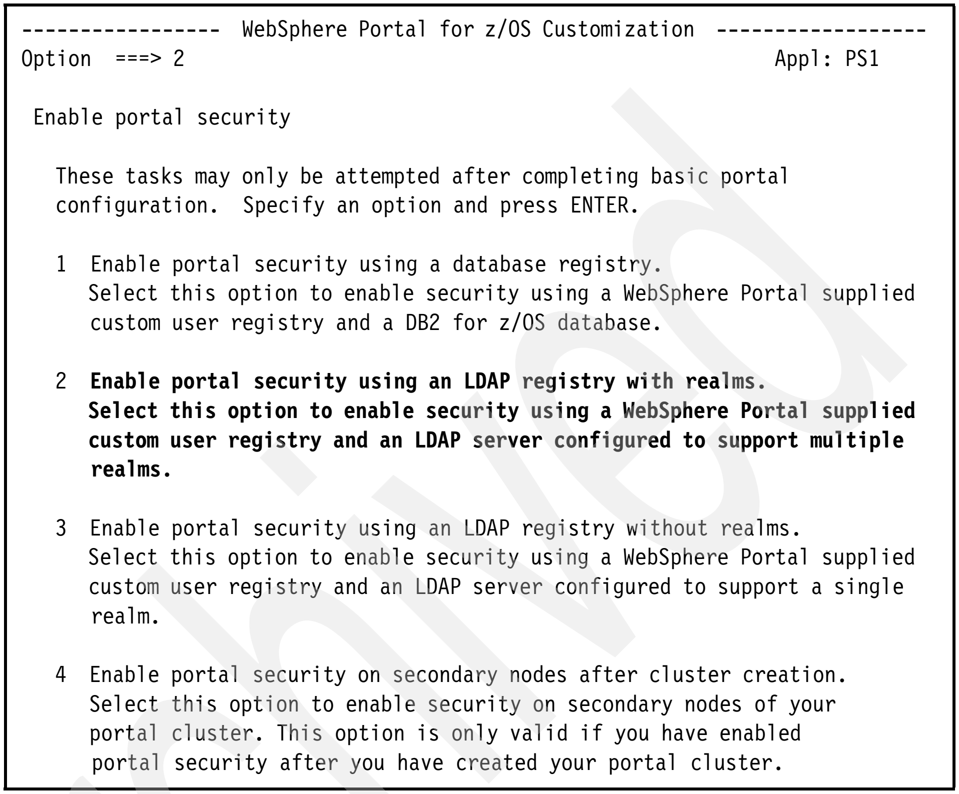
Choose option **1 Enable portal security**, as shown in Figure 4-4.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                     Appl: PS1

 Security configuration tasks

   Use this dialog to specify security options for your portal.
   These tasks may only be attempted after completing basic WebSphere
   Portal configuration. Specify an option and press ENTER.


   1  Enable portal security. Select this option to enable security for your
      portal.  This option is only valid if security is currently disabled
      for your portal.

   2  Disable portal security. Select this option to disable security for
      your portal.  This option is only valid if security is currently
      enabled for your portal.

   3  Configure external security. Select this option to configure your
      portal to use third-party Security authentication and authorization
      servers such as Tivoli WebSEAL Proxy and Tivoli Access Manager.

   4  Change passwords. Select this option to change the passwords for
      the WebSphere Application Server administrator, and the LDAP
      server administrative user.
```

Figure 4-4 Customization panel - enable Portal security

In the next panel, shown in Figure 4-5, select option **2**. We choose to enable security with realm support so that we could create virtual portals later.



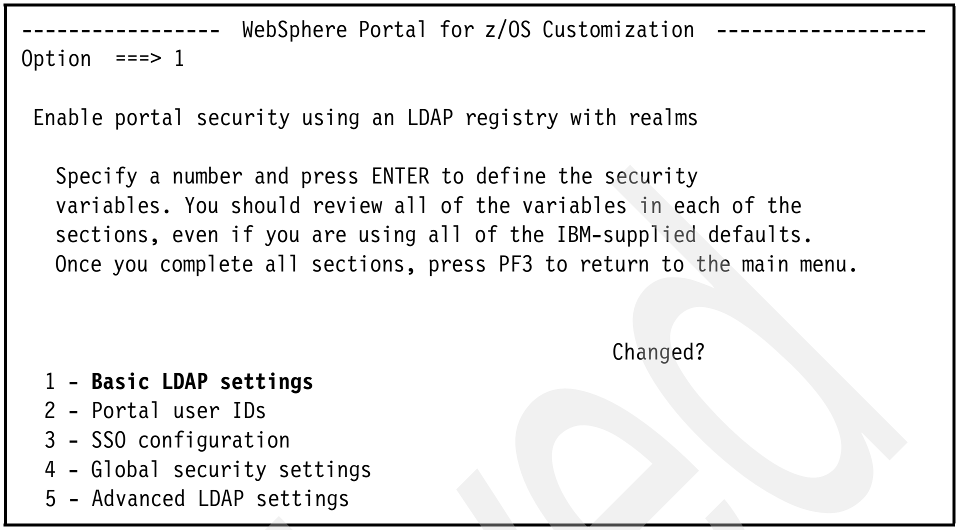
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                   Appl: PS1

 Enable portal security

   These tasks may only be attempted after completing basic portal
   configuration.  Specify an option and press ENTER.

   1  Enable portal security using a database registry.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and a DB2 for z/OS database.

   2  Enable portal security using an LDAP registry with realms.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and an LDAP server configured to support multiple
      realms.

   3  Enable portal security using an LDAP registry without realms.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and an LDAP server configured to support a single
      realm.

   4  Enable portal security on secondary nodes after cluster creation.
      Select this option to enable security on secondary nodes of your
      portal cluster. This option is only valid if you have enabled
      portal security after you have created your portal cluster.
```

Figure 4-5 Customization - enable Portal security using an LDAP registry with realms

Next another panel appears, displaying several options. Use option **L** to load the latest saved variables; these should be the variables saved when you configured the database transfer.

Also specify the data set where the customization jobs are kept. In our case, we kept all customization jobs in the same data set.

Finally, select option **2 Define variables**. This will bring up the next panels, where you will enter information about the LDAP server being used, as shown in Figure 4-6 on page 131.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings
  2 - Portal user IDs
  3 - SSO configuration
  4 - Global security settings
  5 - Advanced LDAP settings
```

Figure 4-6 Customization - selecting basic LDAP settings

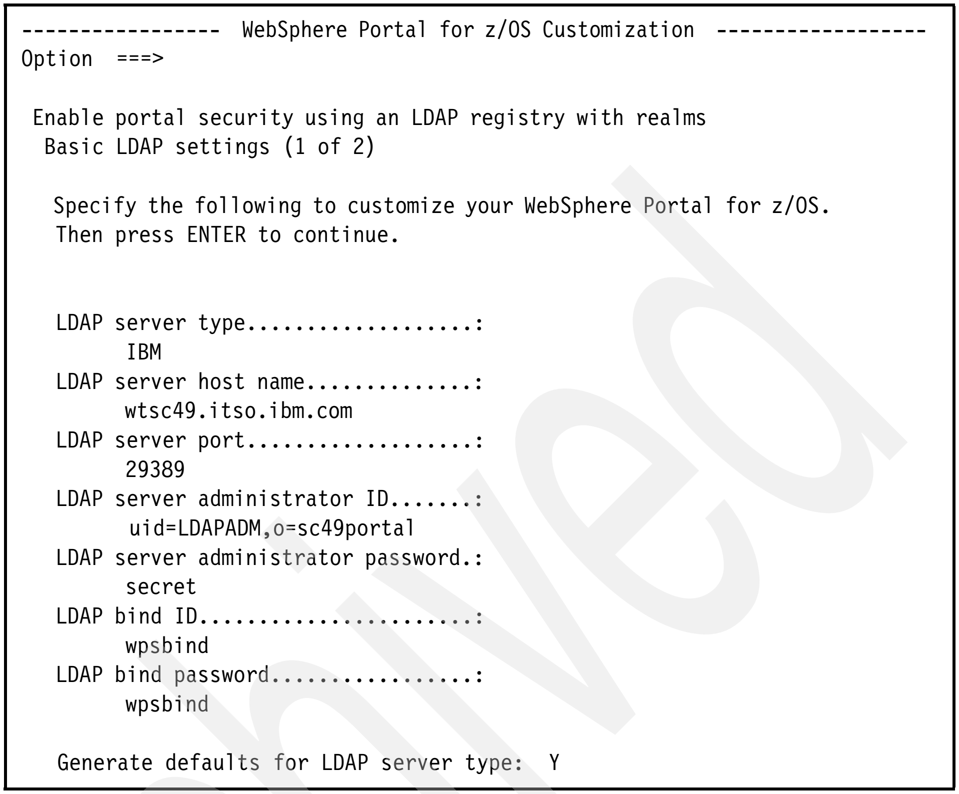
Start with the basic LDAP settings (option 1), which leads you to the panel shown in Figure 4-7 on page 132.

In our case, we used the following values:

- LDAP server type: with an IBM LDAP server, the type is `IBM`.
- LDAP server host name: the host name of our LDAP server
- LDAP server port: the portnumber of our LDAP server
- LDAP server administrator ID: this is the fully qualified name for LDAPADM.

  See Figure 4-1 on page 126 for the location of LDAPADM in the hierarchy.
- LDAP server administrator password: `secret`.
- LDAP bind ID: `wpsbind`.
- LDAP bind password: `wpsbind`.
- Generate defaults for LDAP server type determines if new defaults for subsequent panels will be generated based on this panel. The first time you go into this dialog, choose `Y`. This will cause default values on subsequent panels to be set according to the type of LDAP server you have created.

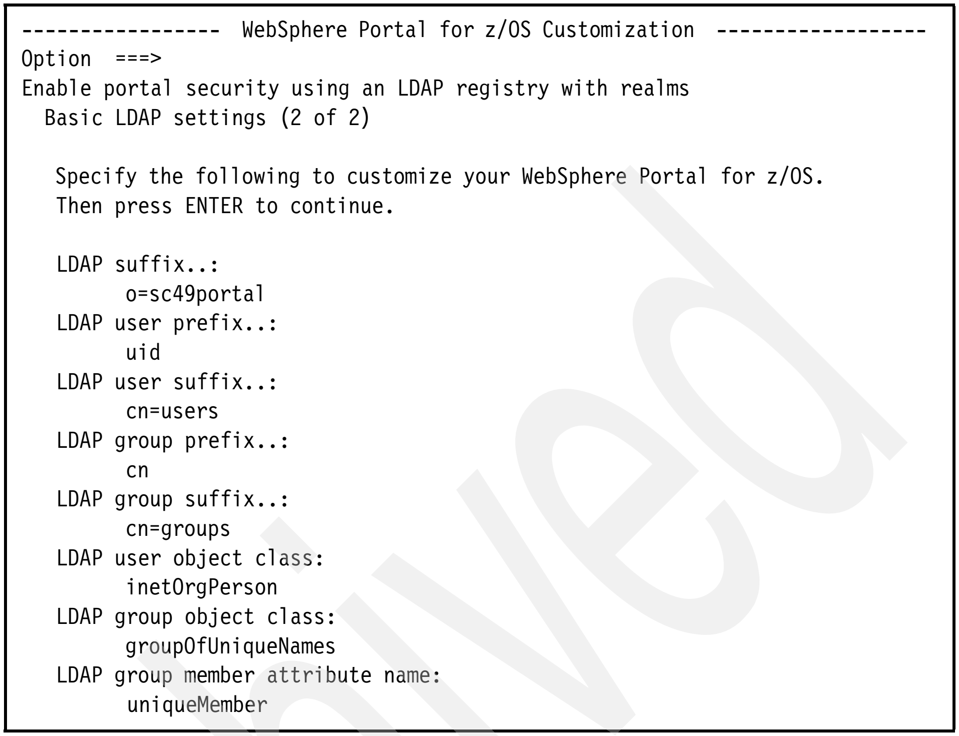
  If you make manual changes on subsequent panels (rather than taking the default), however, those changes will be lost if you reenter the ISPF dialog and set this field to Y. Therefore, choose `N` if you reenter this dialog after configuring Portal security.



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

 Enable portal security using an LDAP registry with realms
  Basic LDAP settings (1 of 2)

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   LDAP server type...................:
         IBM
   LDAP server host name..............:
         wtsc49.itso.ibm.com
   LDAP server port...................:
         29389
   LDAP server administrator ID.......:
         uid=LDAPADM,o=sc49portal
   LDAP server administrator password.:
         secret
   LDAP bind ID.......................:
         wpsbind
   LDAP bind password.................:
         wpsbind

   Generate defaults for LDAP server type:  Y
```

Figure 4-7 Customization - basic LDAP settings

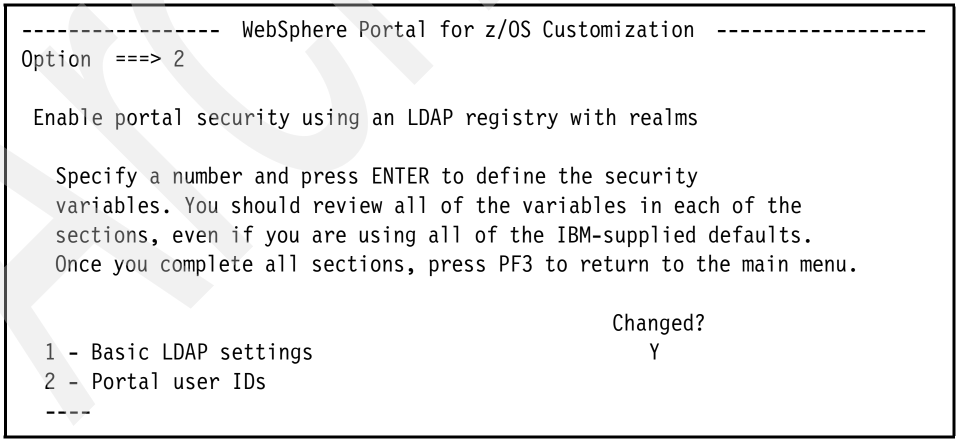
Since in our case we choose to generate defaults for LDAP server type, the only value that must be changed on panel 2 of 2 is LDAP suffix; see Figure 4-8 on page 133. LDAP suffix should be set to the name of the value at the top of the hierarchy.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>
Enable portal security using an LDAP registry with realms
  Basic LDAP settings (2 of 2)

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.

   LDAP suffix..:
         o=sc49portal
   LDAP user prefix..:
         uid
   LDAP user suffix..:
         cn=users
   LDAP group prefix..:
         cn
   LDAP group suffix..:
         cn=groups
   LDAP user object class:
         inetOrgPerson
   LDAP group object class:
         groupOfUniqueNames
   LDAP group member attribute name:
         uniqueMember
```

Figure 4-8   Customization - basic LDAP settings (2 of 2)

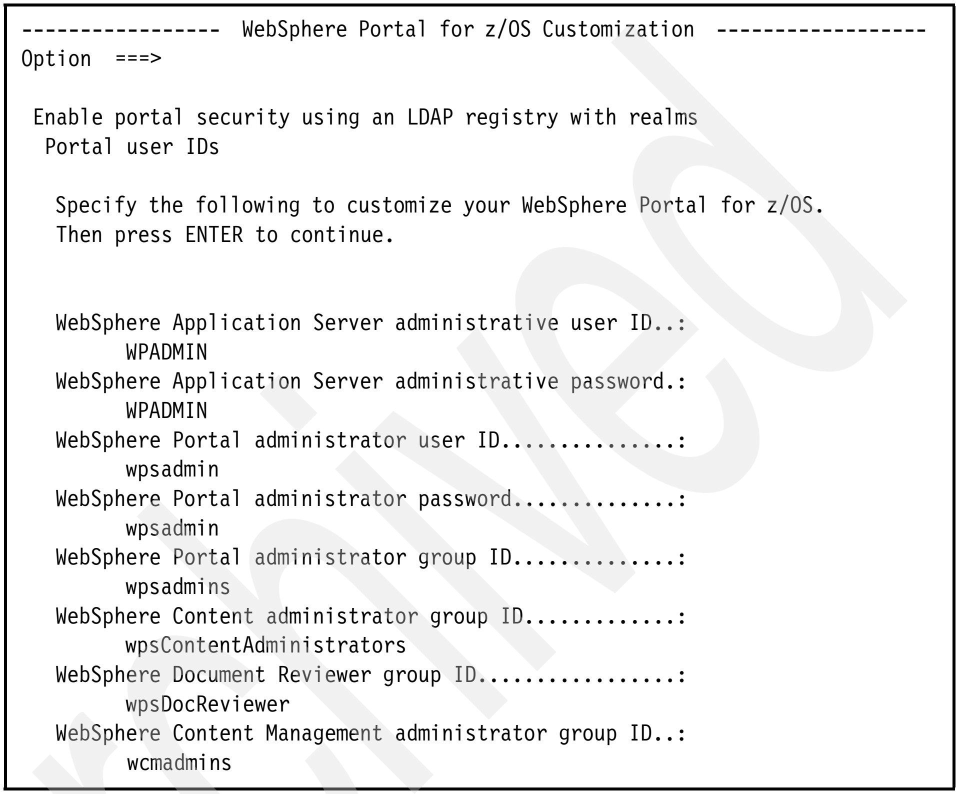
In the next panel, shown in Figure 4-9, enter option 2 for Portal user IDs..



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings                             Y
  2 - Portal user IDs
  ----
```

Figure 4-9   Customization - selecting Portal user IDs

The Portal user IDs panel prompts for all the IDs and groups that will manage the Portal. All of the IDs and groups listed here must exist in LDAP. The document reviewer group and content management administrators are optional and do not have to be specified if content management will not be utilized; see Figure 4-10.

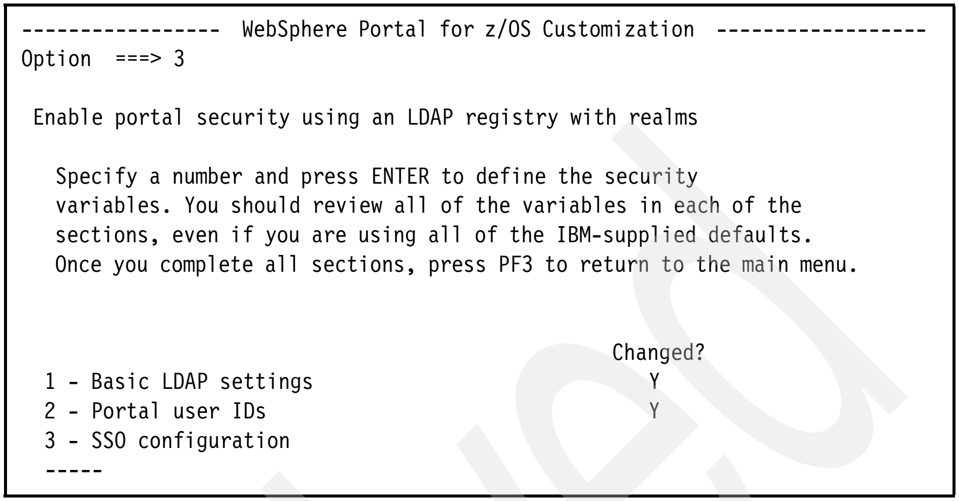


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Enable portal security using an LDAP registry with realms
  Portal user IDs

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   WebSphere Application Server administrative user ID..:
         WPADMIN
   WebSphere Application Server administrative password.:
         WPADMIN
   WebSphere Portal administrator user ID..............:
         wpsadmin
   WebSphere Portal administrator password.............:
         wpsadmin
   WebSphere Portal administrator group ID.............:
         wpsadmins
   WebSphere Content administrator group ID............:
         wpsContentAdministrators
   WebSphere Document Reviewer group ID................:
         wpsDocReviewer
   WebSphere Content Management administrator group ID..:
         wcmadmins
```

Figure 4-10   Customization - specifying user IDs

Remember that passwords in LDAP are case-sensitive.

Next choose option 3, SSO configuration in the panel shown in Figure 4-11.

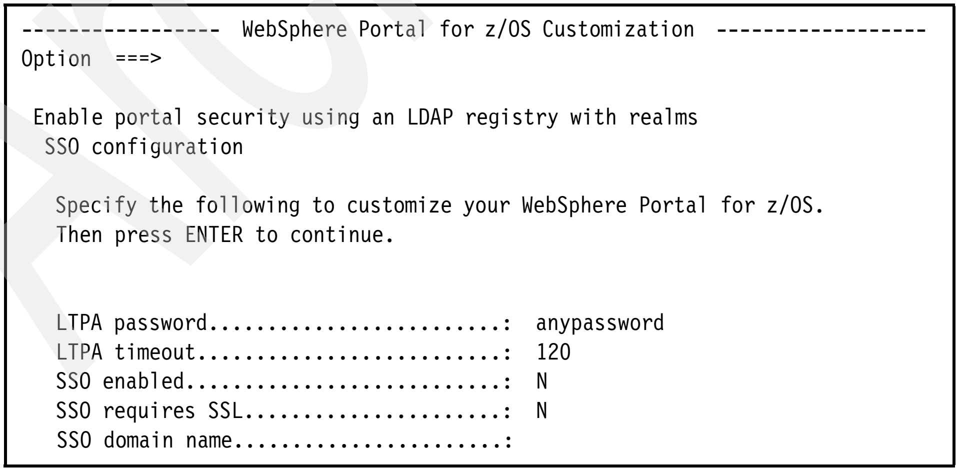


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 3

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings                          Y
  2 - Portal user IDs                              Y
  3 - SSO configuration
  -----
```

Figure 4-11 Customization - SSO configuration

In the panel shown in Figure 4-12:

- Lightweight Third Party Authentication (LTPA) password can be any alphanumeric value and can be changed via the WebSphere administrative console if necessary.
- We choose the default value of 120 minutes for the LTPA timeout. `LTPA timeout` is the number of minutes it takes for an LTPA token to expire.
- SSO was not enabled during our basic security setup.

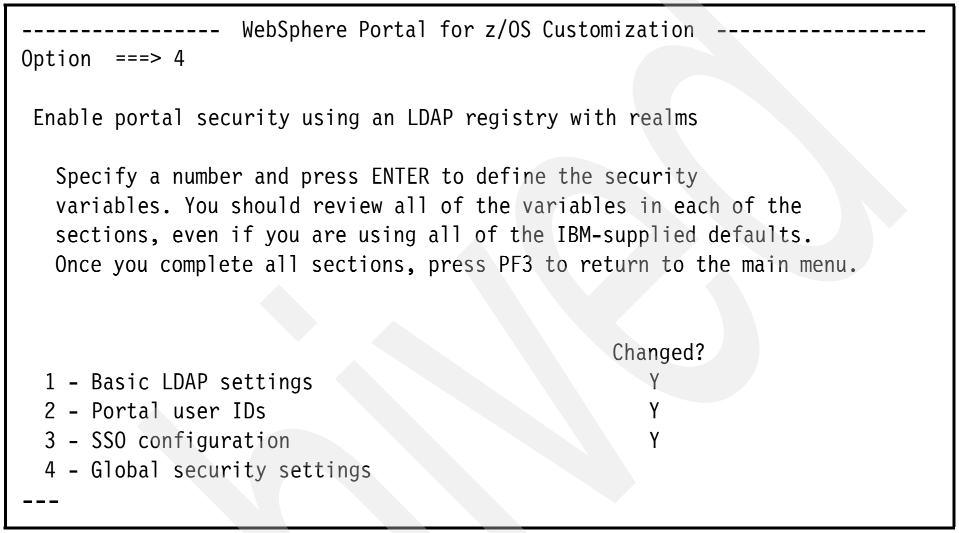


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Enable portal security using an LDAP registry with realms
  SSO configuration

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   LTPA password.........................:  anypassword
   LTPA timeout..........................:  120
   SSO enabled...........................:  N
   SSO requires SSL......................:  N
   SSO domain name.......................:
```

Figure 4-12 Customization - SSO configuration, password

The SSO domain name is not required but if you intend the Portal to be part of a larger single sign-on domain, perhaps involving Tivoli Access Manager and a Domino server, then you need to specify the SSO Domain that all these servers will share.

In the next panel, shown in Figure 4-13, choose option **4 Global security settings**.



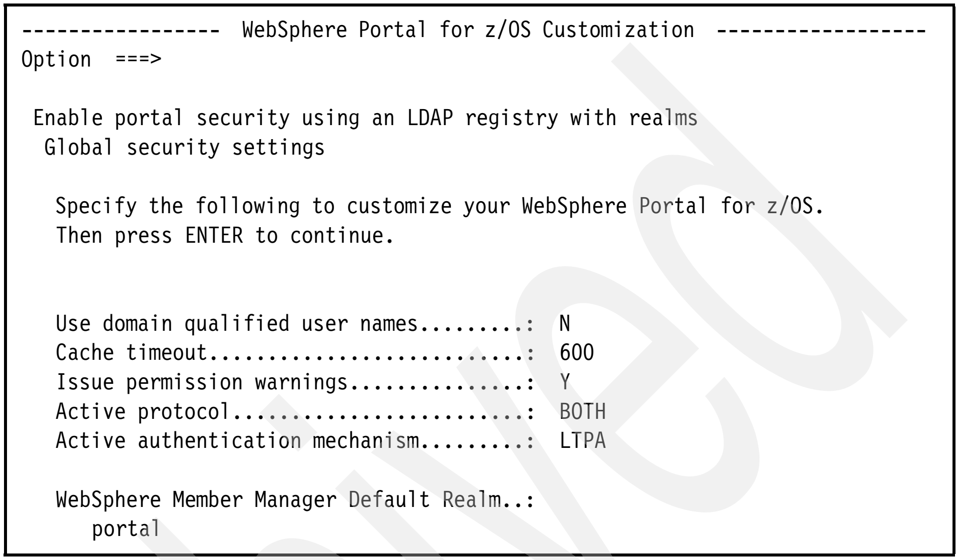
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 4

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings                            Y
  2 - Portal user IDs                                Y
  3 - SSO configuration                              Y
  4 - Global security settings
---
```

Figure 4-13 Customization - selecting Global security settings

Referring to the panel shown in Figure 4-14 on page 137:

- Setting `Use domain qualified user names` to `Y` (for yes) forces the WebSphere administrative console to display user names with their fully qualified domain attributes.
- `Cache timeout` is the number of seconds that authentication information remains in the system cache. After this time, the information is lost and the ID is automatically logged out of WebSphere Application Server.
- With `Issue permission warnings` set to `Y` (for yes), WebSphere Application Server will look at the filter.policy file to see if there are any permissions that a program should not have according to the J2EE 1.4 specification. A warning is issued if a program has authority it should not have.
- A value of `BOTH` for the PROTOCOL means that WebSphere Application Server will use both Common Secure Interoperability (CSI) and the SAS (deprecated) protocol for secure communication.

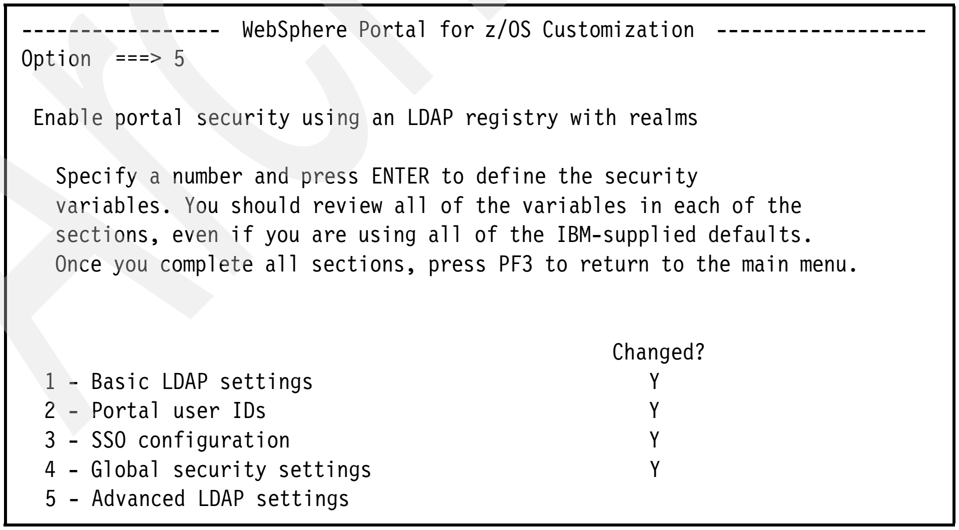
- LTPA is the only option for authentication mechanisms when using WebSphere Application Server Network Deployment.
- Specify the name of the realm.



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

 Enable portal security using an LDAP registry with realms
  Global security settings

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   Use domain qualified user names.........:  N
   Cache timeout...........................:  600
   Issue permission warnings...............:  Y
   Active protocol.........................:  BOTH
   Active authentication mechanism.........:  LTPA

   WebSphere Member Manager Default Realm..:
      portal
```

Figure 4-14 Customization - Global security settings (continued)

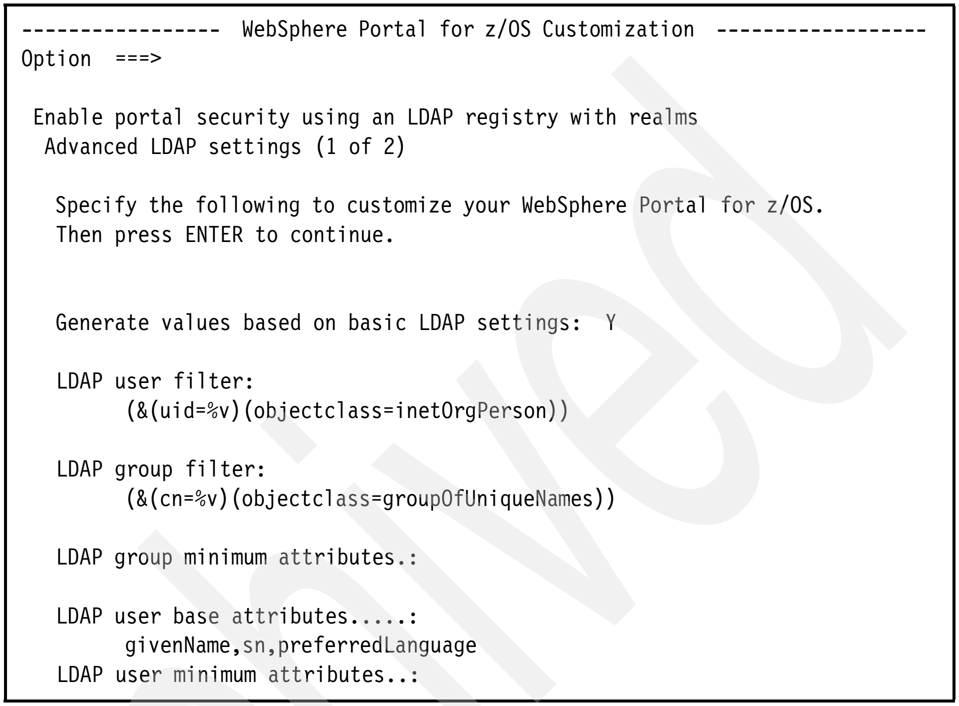
Select option **5 Advanced LDAP settings** in the panel shown in Figure 4-15.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 5

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings                           Y
  2 - Portal user IDs                               Y
  3 - SSO configuration                             Y
  4 - Global security settings                      Y
  5 - Advanced LDAP settings
```

Figure 4-15 WebSphere Portal customization - selecting Advanced LDAP settings

In the panel shown in Figure 4-16, none of the values on the panel were changed since in our case we generated the basic values based on the basic LDAP settings.



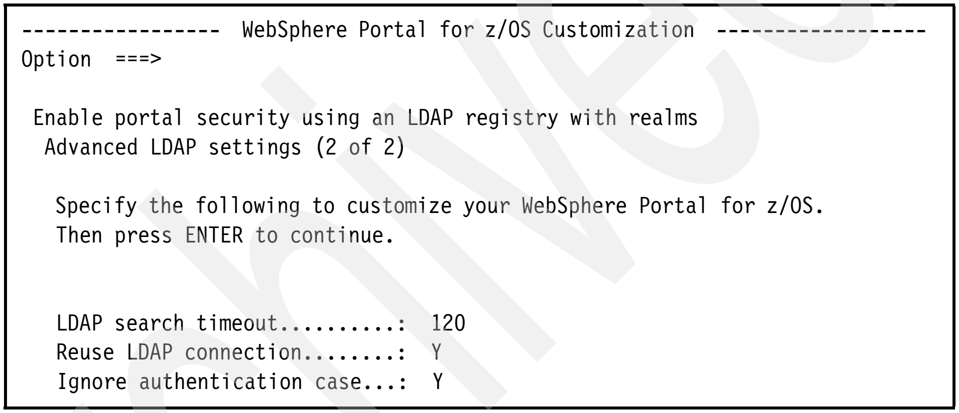
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Enable portal security using an LDAP registry with realms
  Advanced LDAP settings (1 of 2)

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   Generate values based on basic LDAP settings:  Y

   LDAP user filter:
         (&(uid=%v)(objectclass=inetOrgPerson))

   LDAP group filter:
         (&(cn=%v)(objectclass=groupOfUniqueNames))

   LDAP group minimum attributes.:

   LDAP user base attributes.....:
         givenName,sn,preferredLanguage
   LDAP user minimum attributes..:
```

Figure 4-16 WebSphere Portal customization - Advanced LDAP settings (1 of 2)

In the next panel, shown in Figure 4-17:

- `LDAP search timeout` is the number of seconds that WebSphere Application Server will search the LDAP server for an ID before timing out the connection.
- `Reuse LDAP connection` tells WebSphere Application Server to reuse open connections to the LDAP server rather than creating a new connection each time.

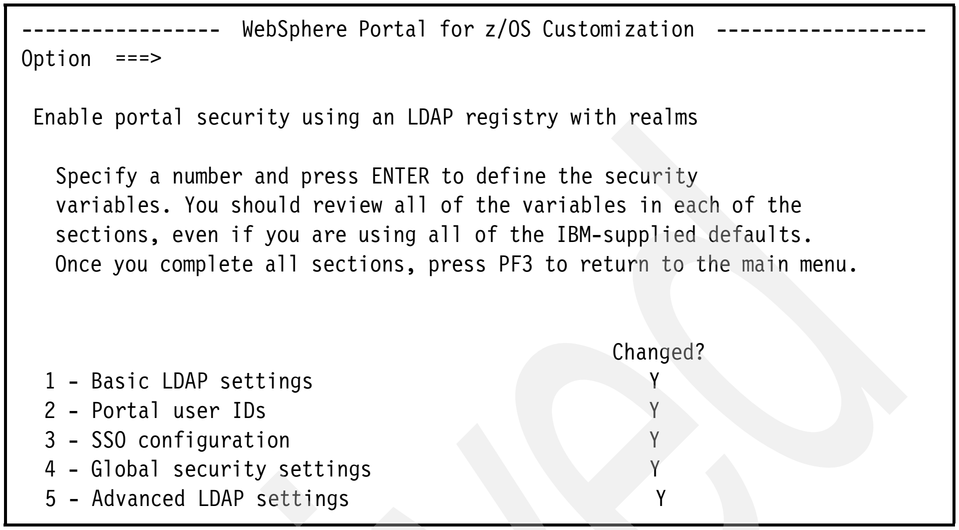
  Note that reusing connections to LDAP has performance benefits, but should not be used in a clustered environment because the Sysplex Distributor does not support system affinity. For more information about this topic, refer to 6.3.2, “Portal considerations with LDAP in multi-server mode” on page 219.
- Set `Ignore authentication case` to `Y` if user IDs and password should *not* be case-sensitive.



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

 Enable portal security using an LDAP registry with realms
  Advanced LDAP settings (2 of 2)

   Specify the following to customize your WebSphere Portal for z/OS.
   Then press ENTER to continue.


   LDAP search timeout..........:  120
   Reuse LDAP connection........:  Y
   Ignore authentication case...:  Y
```

Figure 4-17   Customization - Advanced LDAP settings (2 of 2)

All security parameters have now been updated, as shown in Figure 4-18.



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

 Enable portal security using an LDAP registry with realms

   Specify a number and press ENTER to define the security
   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 - Basic LDAP settings                              Y
  2 - Portal user IDs                                  Y
  3 - SSO configuration                                Y
  4 - Global security settings                         Y
  5 - Advanced LDAP settings                            Y
```

Figure 4-18   Customization - security changed

Press F3 and then select option **3** to generate custom jobs. Choose option **1** to only generate the jobs for this task.

Verify the job card on this screen and press Enter to create the jobs. The messages shown in Example 4-9 will be displayed.

Example 4-9   Security job generation

```
Processing for data set 'WPCELL.PORTALA.CNTL' ...

Member EJPISER successfully created.
Member EJPSSERV successfully created.
Member EJPSSEWL successfully created.
Member EJPSSER successfully created.
Member EJPSCHOU successfully created.
Member EJPSCHIN successfully created.

Processing for data set 'WPCELL.PORTALA.DATA' ...

Member EJP2SER successfully created.

***
```

4.4.1 Running the security jobs

Stop WebSphere Portal before running any of these jobs.

1. EJPSSERV

   This job validates the connection to the LDAP server. It should end with return code of zero (0).

> **Tip:** LDAP passwords are case-sensitive. If you have the wrong password for any ID, you will receive an error message as shown in Example 4-10.

Example 4-10 LDAP connection error message

```
Ýldapcheck¨
###############################################################
Ýldapcheck¨ ldapURL      : wtsc49.itso.ibm.com:29389
Ýldapcheck¨ ldapUser     : uid=WPADMIN,cn=users,o=sc49portal
Ýldapcheck¨ ldapPassword   : *******
Ýldapcheck¨ ldapSslEnabled : false
Ýldapcheck¨
###############################################################
Ýldapcheck¨ javax.naming.AuthenticationException: ÝLDAP: error code 49
- R004062
Ýldapcheck¨ ERROR: 4
Ýldapcheck¨ Invalid or insufficient authorization privileges.
```

2. EJPSSEWL

   This job creates symbolic links in non-Portal nodes. In our case, we only have one non-Portal node (dmgr), but other systems may have several nodes. Make sure this job is run with an ID that has authority to create symbolic links.

   If you do not have Process Choreographer installed, you will have to create a “ProcessChoreographer” directory in the Deployment Manager home directory in order for your job to run successfully.

   ```
   /waswpconfig/wpcell/wpdmnode/DeploymentManager/ProcessChoreographer/
   ```

   Stop and start the Node Agent and Deployment Manager to pick up the change made to the Deployment Manager directory

3. EJPSSER

   This is the job that enables security in Portal and enables Global Security in WebSphere. EJPSSER must be run with the WebSphere Application Server administrator ID, and that ID must have an OMVS segment. This job runs for some time. Times will vary based on the system speed.

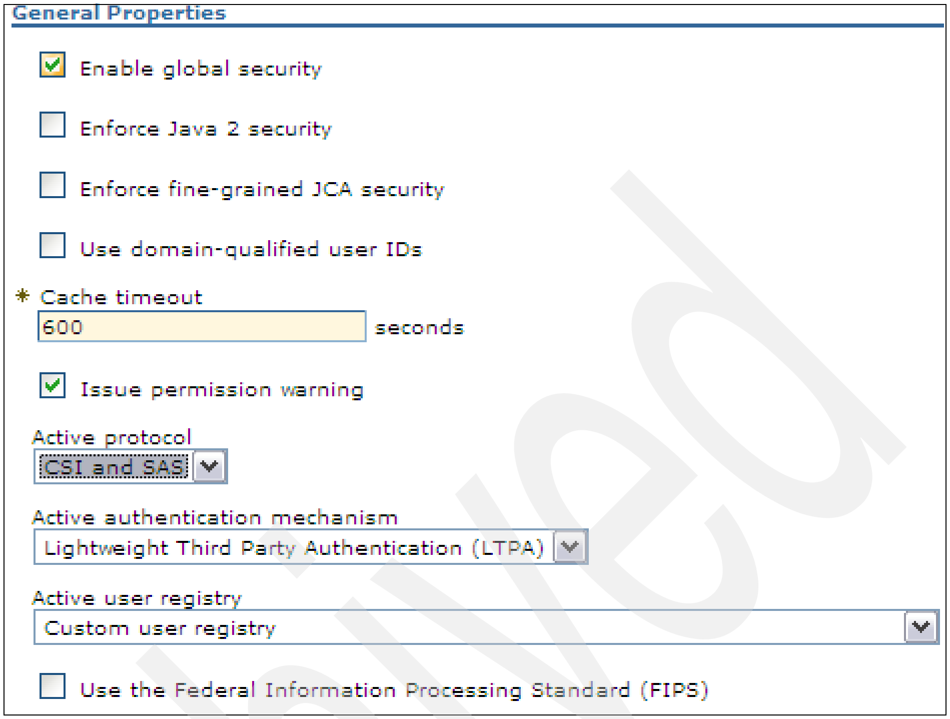
After EJPSSER finishes, perform these tasks:

- Recycle the Deployment Manager and Node Agent.
- Start up the Portal.

  A user logging on to WebSphere Portal should now be able to use the Portal administrator ID in LDAP.
- Log on to the WebSphere Administrative console; it will also use an LDAP ID.

> **Note:** When configuring Portal security with realms, a WMM custom login module is invoked for all authentication requests including those that take place when the Deployment Manager initializes.
>
> WMM uses the cell's WebSphere Administrator user ID and password to connect to the LDAP server. Therefore, if you defined the Administrator's user ID to LDAP with an upper case password but entered a lower case password on the Portal ISPF dialog, then the Deployment Manager will not be able to complete initialization. You will see errors in the log relating to authentication failures for the WebSphere Administration user ID.

- Examine and verify the security enablement using the WebSphere admin console.
  - Select **Security** → **Global security**. The “Enable global security” checkbox should be filled in. The Active authentication mechanism will be shown as `Lightweight Third Party Authentication (LTPA` , and the Active user registry will be shown as `Custom user registry`; see Figure 4-19 on page 143.


General Properties

☑ Enable global security

□ Enforce Java 2 security

□ Enforce fine-grained JCA security

□ Use domain-qualified user IDs

* Cache timeout
600 seconds

☑ Issue permission warning

Active protocol
CSI and SAS

Active authentication mechanism
Lightweight Third Party Authentication (LTPA)

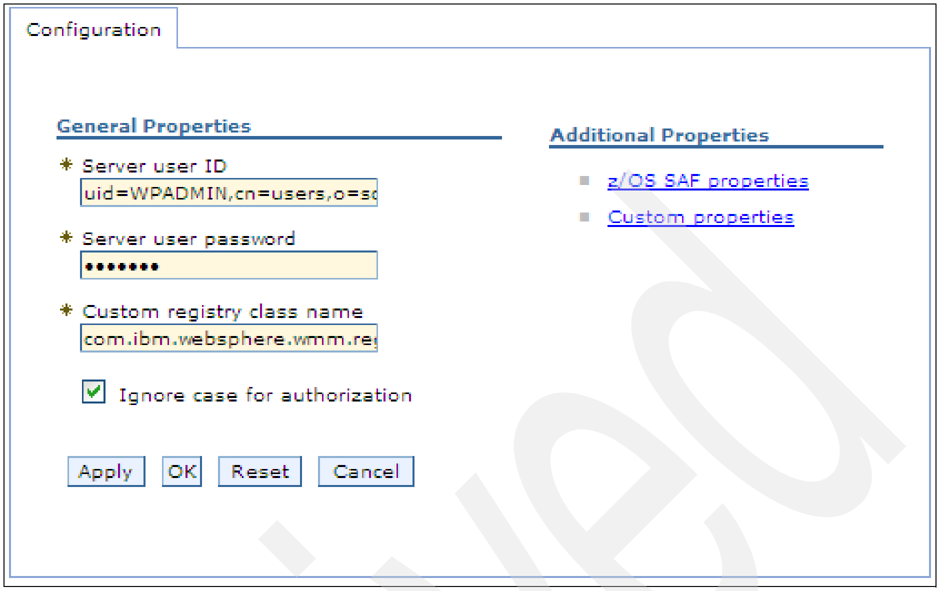
Active user registry
Custom user registry

□ Use the Federal Information Processing Standard (FIPS)

Figure 4-19 Security > global security

In our case, the Active user registry is a Custom user registry because we defined a genuine LDAP directory.

– Next, select **Security** -> **Global security** -> **Custom user registry**. The user ID that was given as the WebSphere Application Server administrator will be in the server user ID field; see Figure 4-20 on page 144.



* Server user ID: uid=WPADMIN,cn=users,o=sc

* Server user password: •••••••

* Custom registry class name: com.ibm.websphere.wmm.re

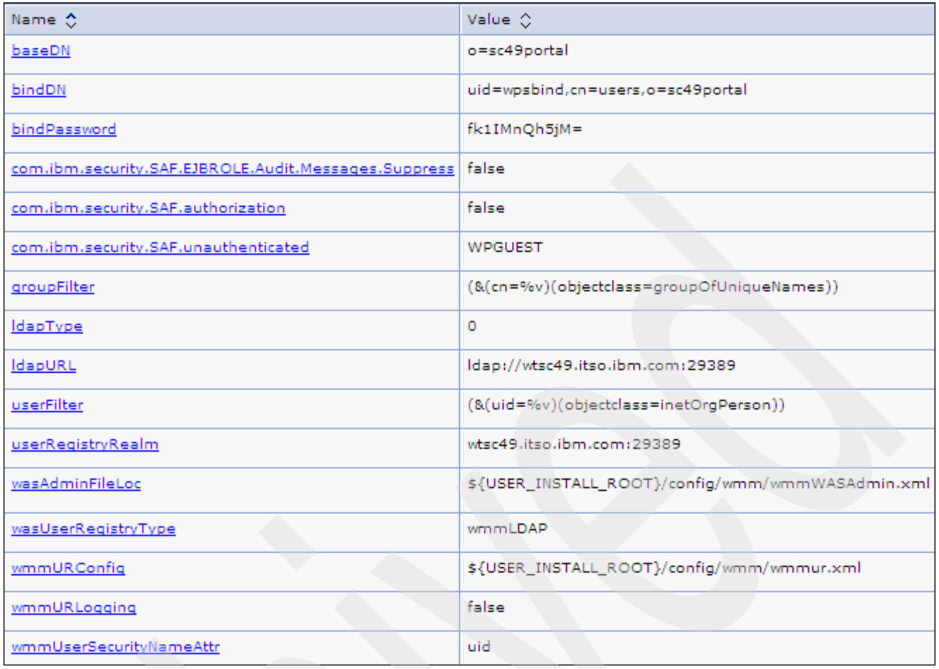
☑ Ignore case for authorization

Figure 4-20 Security > Global security > Custom user registry

- Select **Security** -> **Global security** -> **Custom user registry** -> **Custom properties**.

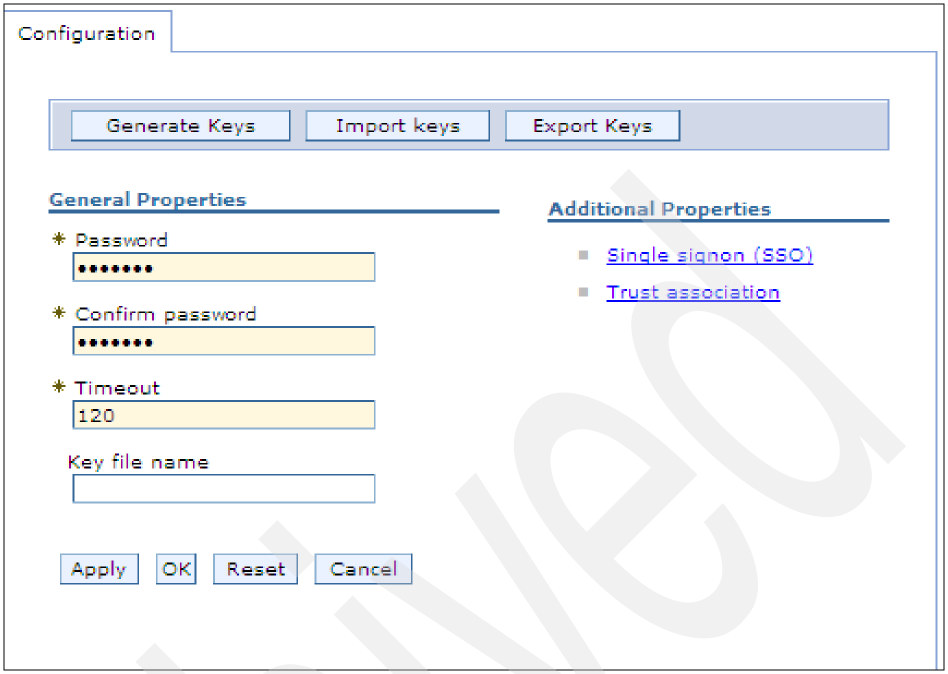
  Many of the variables that were entered into the WebSphere installation dialog are now defined on this screen as custom properties; see Table 4-1 on page 145.

Table 4-1 Security -> Global security -> Custom user registry -> Custom properties



Name	Value
baseDN	o=sc49portal
bindDN	uid=wpsbind,cn=users,o=sc49portal
bindPassword	fk1IMnQh5jM=
com.ibm.security.SAF.EJBROLE.Audit.Messages.Suppress	false
com.ibm.security.SAF.authorization	false
com.ibm.security.SAF.unauthenticated	WPGUEST
groupFilter	(&(cn=%v)(objectclass=groupOfUniqueNames))
ldapType	0
ldapURL	ldap://wtsc49.itso.ibm.com:29389
userFilter	(&(uid=%v)(objectclass=inetOrgPerson))
userRegistryRealm	wtsc49.itso.ibm.com:29389
wasAdminFileLoc	${USER_INSTALL_ROOT}/config/wmm/wmmWASAdmin.xml
wasUserRegistryType	wmmLDAP
wmmURConfig	${USER_INSTALL_ROOT}/config/wmm/wmmur.xml
wmmURLogging	false
wmmUserSecurityNameAttr	uid

- Finally, select **Security** -> **Global Security** -> **Authentication Mechanisms** -> **LTPA**. This is where the LTPA keys are defined; see Figure 4-21 on page 146.



General Properties

* Password: •••••••

* Confirm password: •••••••

* Timeout: 120

Key file name:

Additional Properties

- Single signon (SSO)
- Trust association

Figure 4-21 Security > Global Security > Authentication Mechanisms > LTPA

5

High availability scenarios

There are many factors involved in the process of creating a high availability environment, including hardware, software, environmental considerations, monetary considerations, risk assessment, culture, and personnel resources. In this chapter we explore these and other considerations as we present a methodology for creating and maintaining a Portal high availability infrastructure.

We include problem descriptions and use cases for three major areas of availability and recovery:

Planned outages These are outages that are well understood and planned for in advance. The reasons for these outages may include software or hardware upgrades, personnel issues, facilities tests, and so on.

Unplanned outages These are outages that are not planned for in advance. This category includes all software and hardware-related problems which cause application unavailability.

Disaster recovery These are outages that result in the catastrophic loss of data, materials, infrastructure, or personnel. This is a special category of outage that must be dealt with specifically for mission-critical applications.

5.1 A general discussion of high availability

In the following sections we present a general discussion of high availability.

5.1.1 What is high availability

When an organization acquires a new computer system, it is investing resources (both financial and human) in a new asset. This asset requires care and attention in order to produce its maximum return on investment. A key attribute influencing the level of return that a system can provide is its level of availability. *Availability* is the proportion of time that a system is able to be used for its intended purpose.

Availability has become a significant issue for many enterprises today. With computerized applications becoming more critical to business operations, the extent to which companies rely on their computer systems has never been greater. The amount of availability that a system provides is dependent on a range of issues, many of which we will discuss.

5.1.2 Why availability is important

Computer systems are a critical part of most businesses. For example, it is difficult to imagine reserving a seat for a flight or withdrawing money from your bank account at 2:00 a.m. without the support of computer systems. Once a level of computer system function is available, then its consistent availability becomes a key contributor to the success of the business—and any reduction of system availability will incur costs (direct or indirect) to the business.

Before you develop an availability strategy, you first need to understand the value of availability and the cost of reduced availability of each specific application to the business. Then use this knowledge to make judgments as to the amount of resource to invest in availability solutions. Inevitably, tradeoffs will be made between the cost of the solution and the cost of an outage. This can only be done effectively if you understand each system's value to the business, as well as the impact of its loss over time.

If you have decided to use a Portal as the front-end for your business applications, this implies that the availability requirement of the Portal server is greater than that of any individual application. To put it another way, if your Portal is down, everything behind that Portal is down. That is why choosing Portal on z/OS is such a good decision for organizations that require 24x7 operation of their applications.

5.1.3 Levels of availability

Availability can be seen as a continuum, where each point along the axis has an associated cost, and where improvements can be obtained through investment in additional resources and technologies. Computer vendors seek to ensure that the systems they install have reliable hardware and software to provide reasonable levels of availability without additional investment. In general, for a given system, the greater the level of availability desired, the greater the cost of achieving it.

There are four basics levels of availability: basic, improved, high, and continuous. We explore these levels in more detail in the following sections.

Basic availability

Basic availability is the level of availability achieved with a single system and with basic systems management practices in place. For many installations and systems, this is a sufficient level of availability. Note that basic systems management is required to maintain this level of availability.

Normal operational cycles include planned system outages for system management and application changes. But it should also be expected that unplanned outages will occur. For this reason, basic recovery procedures must be in place. Expected recovery times from unplanned outages can vary widely, from a few minutes for a reboot caused by a simple configuration change, to many hours or days for severe hardware, software, or environmental issues.

Selecting reliable hardware can be a factor in determining how reliable a system is likely to be. When planning for good basic availability, evaluate the underlying hardware as well as any software components that work in conjunction with the hardware. Good customer support and responsiveness are also critical in achieving an acceptable level of basic availability.

Improved availability

Systems with *improved availability* provide greater robustness through the application of additional technology and resources to one or more system components. This additional technology provides greater availability at a greater cost than a similarly configured system with basic availability.

Techniques such as disk mirroring, the use of an uninterruptable power supply (UPS), redundant components, data journaling, data forwarding, hot pluggable disk drives, and disk sharing can each be used to address specific problem areas. For example, a system that exploits mirrored disk arrays will more easily overcome a failure caused by a disk failure.

The goal of improved availability is not to provide continuous, uninterruptable service—it is to provide the means by which you can recover a system from an unplanned outage. These systems must be *designed* for higher availability, and more rigorous systems management procedures must be in place to deal with the inevitable unplanned failure. The goal of a system with improved availability is to be able to recover from a failure in anywhere from a few minutes to a few hours.

High availability

Highly available systems attempt to provide continuous service within a particular operational window by minimizing the causes of failure and reducing the time necessary to recover when a failure does occur. Generally, this requires a high degree of redundancy in system components and system functions so that the continued operation of the system is protected from the failure of any one component. The ultimate objective is to reduce or eliminate any single points of failure (SPOF) in the system.

It is a requirement of highly available systems that the recovery time for any unplanned outage be as brief as possible. Highly available systems are still likely to require outages for systems management activities, but these outages will always occur outside of the normal operational window. Recovery times for highly available systems are expected to be on the order of tens or hundredths of a second. Applications such as WebSphere Portal and WebSphere Application Server may experience dropped connections, but they provide the ability for clients to immediately reconnect and recover their sessions. No additional client wait time, queue time, or processing steps are permitted as a consequence of recovering a dropped connection.

Achieving this level of availability requires a significant investment in hardware, software, personnel, and procedures. Strict change management regimens must be put into place and e rigorously adhered to. Automation must be used extensively. And operations staff must be fully trained and drilled frequently on recovery procedures.

Continuous availability

At this level of availability, the system never fails to deliver service and always meets or exceeds its specified service level agreement (SLA). These systems provide 100% availability to their clients by providing both redundancy in components and the ability to perform automated error detection and recovery in the event of a component failure. In this environment outages may occur, but they should never be exposed to clients as a reduction in service.

5.1.4 Factors that may impact high availability

Many factors contribute to a high availability solution, as explained here:

- Hardware

  Although computer systems are becoming more and more reliable, hardware failures still occur. And the hardware failure of a system may affect all or part of a system's function.

  Furthermore, to repair the failed component, the system may need to be shut down completely, either to run diagnostic procedures or to enable replacement of the failed hardware components.

- Software

  The operating system, middleware, and applications may fail for a variety of reasons. Sometimes they fail because of defects, and sometimes they fail because of errors in installation or configuration.

  These failures affect operations to varying degrees. Production time may be lost during the attempt to recreate and isolate the problem, or during the process of applying updates or fixes.

- Environment

  To function properly a system must have adequate power, lighting, cooling, and so on in order to operate within its environmental bounds. For example, if a system has no power, it cannot operate and provide service. If a system is not adequately cooled, the chances of a system failure are greatly increased and reliability will decline.

  The reliability of electrical power systems varies widely from location to location. In some locales, power interruptions may occur as frequently as twenty times per year. Also, many locations have planned outages that may affect production systems.

- Communications

  Communications are a vital part of a system's infrastructure and a frequent source of system availability problems. While the system itself might be adequately protected, most installations rely on a third party provider for network connectivity. Wide area networks (WANs) are subject to a variety of problems, most of which are beyond the control of the end user.

  The role of network availability must be carefully considered by the system designer. The use of network management strategies and products may be necessary to alleviate SPOF, at least to the point of network attachment to the common carrier.

- Systems management and change control

  Poor systems management and change control procedures can contribute to increased system downtime. Unscheduled outages can be caused by seemingly simple items as server reconfigurations, incomplete backups and undocumented system status. A lack of skill in systems management can make establishing a highly available installation problematic.

- Monetary

  When a system and its supported business processes are unavailable, it may be possible to provide an alternative solution simply by providing another complete system. Although it may be tempting to address availability problems in this way, there will most certainly be costs involved. Very few availability problems can be fixed in the long term simply by spending more.

  At the same time, the degree of availability that can be obtained from a system will be driven in large part by the cost of the associated systems and infrastructure necessary to achieve that level of availability.

- Risk

  Availability is ultimately a balanced evaluation of the costs associated with the infrastructure required to achieve the desired level of availability, and the costs and benefits of providing that level of service to clients. Risk of failure and the associated loss of business revenue must be carefully measured against the costs of providing a highly available system.

- Culture

  Highly available systems almost always have a culture of high availability surrounding them. Hardware, software, and their associated technologies all have roles to play in creating a robust, highly available system. People and the culture of their work environment complete the picture. If availability is made a priority in the organization, then training will become important. Automation and systems management procedures will be at the top of the agenda. And management will support the goals of the highly available system and devote the necessary time and resources to achieve it.

5.1.5 Types of outages

There are three different classes of system outages that a robust availability architecture must be prepared to deal with: planned outages, unplanned outages, and disaster recovery (DR) outages. We discuss these in more detail in the following sections.

Planned outages

Under this category we include all facilities and procedures for minimizing downtime associated with planned outages. This includes application migration, server migration, and systems management changes. In reality, this category of problems is less about outages and more about how to minimize downtime so that client application availability can continue. An additional focus of this area is how to avoid a complex-wide outage when applying updates and changes.

Unplanned outages

There are two aspects of this category of outage:

- Because this is the primary problem area for most installations, you need to determine the optimal configuration of servers and software to minimize unplanned interruptions of service.
- While there is an operational component to this problem, the larger question is "Can you enable the various components of the system for automated recovery and for operational procedures to occur?" You need to determine what configuration changes are necessary to enable recovery processing to occur automatically when failures are detected.

Disaster recovery

Disaster recovery (DR) issues can be particularly challenging and involve the partial or complete loss of all data and facilities associated with a computing site. For example, after the events of September 11, 2001, entire data centers had to be recreated almost overnight. In that case, many enterprises were prepared and had disaster recovery procedures in place. For these enterprises, the additional cost of maintaining a redundant backup environment was entirely acceptable.

For the most part, facilities and procedures for disaster recovery are maintained outside of the normal daily operation of the computing center. Therefore, systems can be designed to be highly available and also support good disaster recovery procedures. These two goals do not conflict with one another and may actually enhance each another.

5.2 Possible SPOFs in a WebSphere Portal-based system

A highly available system attempts to reduce the number of single points of failure. In a WebSphere-based system there are many techniques for achieving this goal. Table 5-1 on page 154 lists suggestions about how to accomplish this goal.

Table 5-1 Single points of failure in a WebSphere-based system

Failure point	Possible solutions
Client access	Multiple ISPs
Firewall	Firewall clustering, firewall sprayer, highly available firewall
Caching proxy	Backup caching proxy
HTTP sprayer	Backup sprayer, vendor-specific solution (that is, Cisco MNLB), Sysplex Distributor
Web server	Multiple Web servers utilizing traffic router, network load balancers
WebSphere master repository file, log files	NFS, SAN, data mirroring, shared sysplex HFS
WebSphere Application Server	Application clustering ► Vertical via multiple servants ► Horizontal via multiple sysplex nodes
WebSphere Node Agent	Multiple nodes in the cluster
WebSphere Deployment Manager	Deployment Manager configured so it can be started on any LPAR in the sysplex
Portal	Application clustering ► Vertical via multiple servants ► Horizontal via multiple sysplex nodes
Entity EJB	DB2 data sharing
Portal configuration databases	DB2 data sharing
Application databases	DB2 data sharing
Session database	DB2 data sharing
Transaction logs	► XA logs - NFS, SAN, data mirroring, shared sysplex HFS ► z/OS logs - RRS, Coupling Facility
Messaging provider	MQ shared queues, MQ clusters
Directory	Backup or load balanced directory server, shared RACF database
Local area network	Redundant OSA cards, VIPA, DVIPA
Wide area network	Multiple network provider connections

Failure point	Possible solutions
Disk failures	Disk mirroring, multiple I/O paths, TotalStorage
Network service failures (DNS, DHCP, and so on)	Redundant servers
Operating system error	z/OS, Parallel Sysplex
Server failure	▶ System z hardware - self-healing ▶ Parallel Sysplex ▶ Coupling Facility
Power failure	Dual power drops, UPS, backup power supply
Software upgrades, hardware upgrades	Rolling upgrades, application versioning
Room, floor, location disaster	Parallel Sysplex components in different buildings connected via channel extenders
City disaster	GDPS® solutions - Remote Mirror, replication, PPRC, XRC, HyperSwap, outsourced DR
Regional disaster	GDPS solutions - Remote Mirror, replication, PPRC, XRC, HyperSwap, outsourced DR
Human error	Training, simplified procedures, simplified systems management

5.3 WebSphere Application Server and WebSphere Portal availability scenarios

We can deploy WebSphere Application Server and WebSphere Portal with different levels of redundant hardware, software, networks, processes, and components.

For the purpose of this discussion, we divide the deployment issues into several availability levels (1 through 5), where 1 is basic availability and 5 is very high availability. Subsequent levels include all characteristics of the preceding levels unless otherwise noted.

5.3.1 Notes for HA levels 1 - 4

We make the following assumptions for HA levels 1 - 4:

- A DMZ is not shown, but is assumed to consist of duals firewalls with a HTTP server configured in a reverse proxy configuration to allow inbound traffic to the proxy and not through the inner firewall.
- An LDAP server is configured on z/OS for security.
- All configuration and application data is stored in a DB2 for z/OS V8.1 database.
- Transaction recovery is handled by RRS or the Coupling Facility, or by RRS and the Coupling Facility.
- Log recovery is handled by zFS or HFS, or by zFS and HFS.

5.3.2 WebSphere/Portal HA level 1

For HA level 1, the HTTP server, WebSphere Application Server server, WebSphere Portal and DB2 subsystem are all installed on a single z/OS LPAR; see Figure 5-1. There is one Portal to service client requests. If the LPAR, or any of the LPAR components, fail, then the system is unavailable to its clients.

HA level 1 represents basic availability. Availability is guaranteed as long as the LPAR and all of its system components remain available. The moment any one system component fails, availability is lost. This configuration is good for a development environment where availability is not considered the most important factor in the system design.

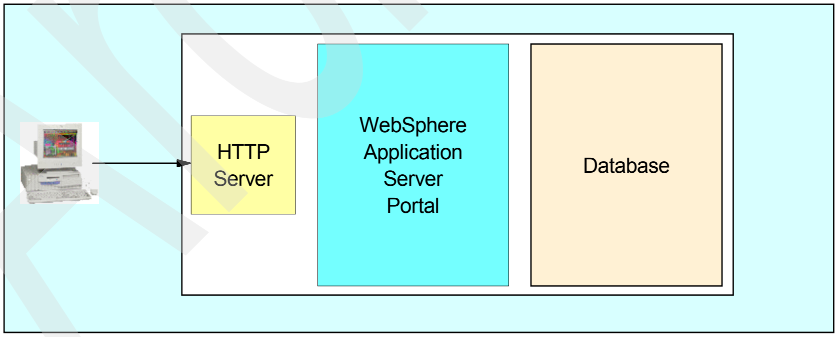


Figure 5-1 WebSphere/Portal system HA level 1

Single points of failure include:

- HTTP server
- Application server
- Portal
- DB2 database (application data)
- Firewalls (not shown)
- Directory (not shown)
- z/OS operating system
- Hardware (includes all hardware components including servers, network, power, cooling, and so on)

5.3.3 WebSphere/Portal HA level 2

A WebSphere z/OS server can comprise multiple servant regions which act like a “mini-cluster” in that work is distributed between them by the zWLM. This gives WebSphere and Portal on z/OS much greater resilience than the same configuration on non-z/OS platforms. Figure 5-2 on page 158 illustrates a WebSphere/Portal system HA level 2.

In the case of a servant failure, then z/OS, WLM, and the WebSphere Control region will restart a new servant. RRS will clean up any transactions for the failed server. Because all servants are on the same LPAR and share a common file structure, no shared HFS is required. The servants also share a common DB2 subsystem.

HA level 2 has enhanced availability. Availability of the WebSphere and Portal servers is reasonably good, but SPOFs still exist in the infrastructure components and the hardware. This environment is suitable for development and for some FVT tests and production applications that are not considered critical for business.

Note: HA level 2, as with all WebSphere deployments on a z/OS LPAR, requires sufficient system resources to host the environment. As more servants are deployed on a single LPAR, more memory and CPU processing power will be required.

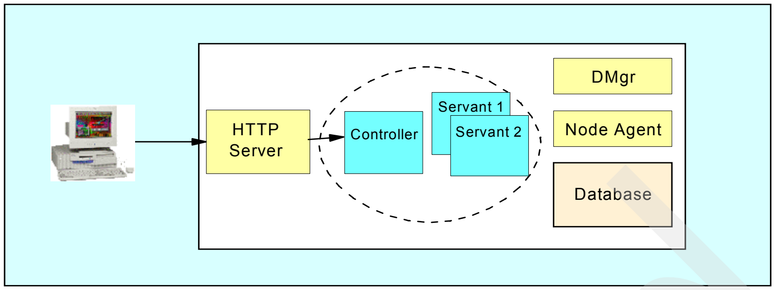


Figure 5-2 WebSphere/Portal system HA level 2

Single points of failure include:

- HTTP server
- Administrative servers (especially Node Agent)
- DB2 (application and session data)
- Firewalls (not shown)
- Directory (not shown)
- z/OS operating system
- Hardware (includes all hardware components including servers, network, power, cooling, and so on)

5.3.4 WebSphere/Portal HA level 3

For HA level 3 a load balancer, multiple HTTP servers, and WebSphere horizontal clustering utilizing a second system in the sysplex hosting WebSphere are all in use. DB2 data is accessed in a separate LPAR via a network connection. If either system in the sysplex fails, then the surviving system can recover in-flight or in-doubt transactions. Figure 5-3 on page 159 illustrates a WebSphere/Portal system HA level 3.

HA level 3 has enhanced availability. Availability of the WebSphere/Portal servers is very good, both in terms of software and in terms of hardware failures. The file systems have been addressed, as have the HTTP servers feeding the LPARs. Significant SPOFs still exist, however, particularly in the DB2 configuration. The supporting servers such as CICS, DB2, CICS, and MQ also need to be configured in a similar way for HA level 3.

HA level 3 systems are well- suited to FVT testing and small production environments where significant outage times are acceptable in the event of a database or infrastructure component failure.

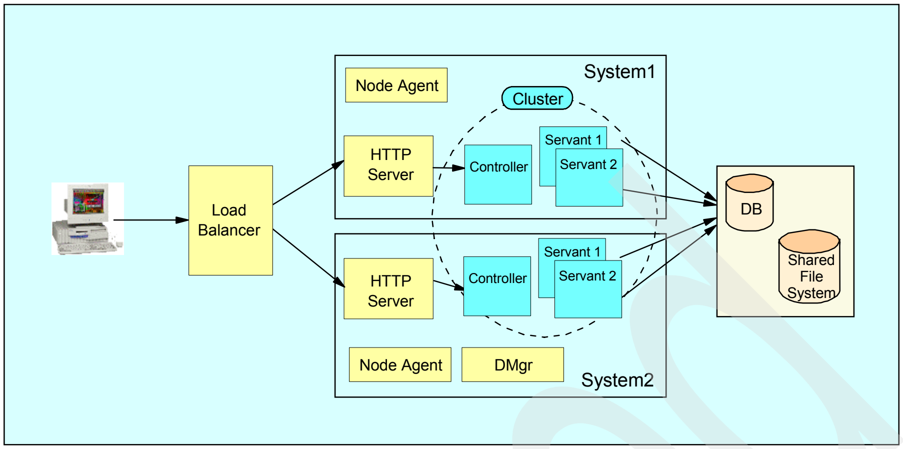


Figure 5-3 WebSphere/Portal system HA level 3

Single points of failure include:

- Load balancer
- Administrative servers
- DB2 (application and session data)
- Firewalls (not shown)
- Directory (not shown)
- z/OS operating system
- Hardware (backup still not provided for some components: hardware components including some servers, network, power, cooling, and so on)

5.3.5 WebSphere/Portal HA level 4

A WebSphere/Portal HA level 4 system eliminates most of the SPOFs which present in lower HA levels.

- There is a backup load balancer.
- There are multiple HTTP servers.
- The combination of vertical and horizontal scaling protects against both software and hardware failures in the WebSphere/Portal infrastructure.
- A shared file system is in use.
- The database is configured with full DB2 data sharing across the sysplex. In the event of a system failure, database locks may be recovered by the surviving member of the sysplex.

The administrative servers (Node Agents and Deployment Manager) are the only SPOFs in this system—but only from a management point of view, not from an installed application's availability point of view.

- A failure of the Deployment Manager is not normally a major problem because running production WebSphere servers do not require it to function.

  Configuration changes will not be possible as long as the Deployment Manager is unavailable. In addition, administrative commands such as start and stop will not be able to be directed against a server cluster (although individual z/OS commands directed against a server will continue to function.)

  Temporary failures of the Deployment Manager can be tolerated in this environment as long as there are procedures in place for restarting it. System automation or use of the z/OS Automatic Restart Manager (ARM) facility are useful ways to ensure the Deployment Manager remains available.

  The Deployment Manager can be configured so it can be started on any LPAR in the sysplex. For more information about this topic, refer to white paper WP100585 "Starting the DMGR on a Different MVS Image" which can be obtained from the following site:

  `http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100585`

  You can configure an ARM policy to restart it automatically in the event of a failure.

- A failure of a Node Agent will prevent configuration changes from being synchronized to the node by the Deployment Manager. However, this does not prevent the applications from functioning.

  When the Node Agent is restarted, any pending configuration changes will be synchronized at that time.WebSphere/Portal HA level 4 supports high availability and is well-suited to all production environments. Only the largest, most mission-critical deployments need a higher level of availability.

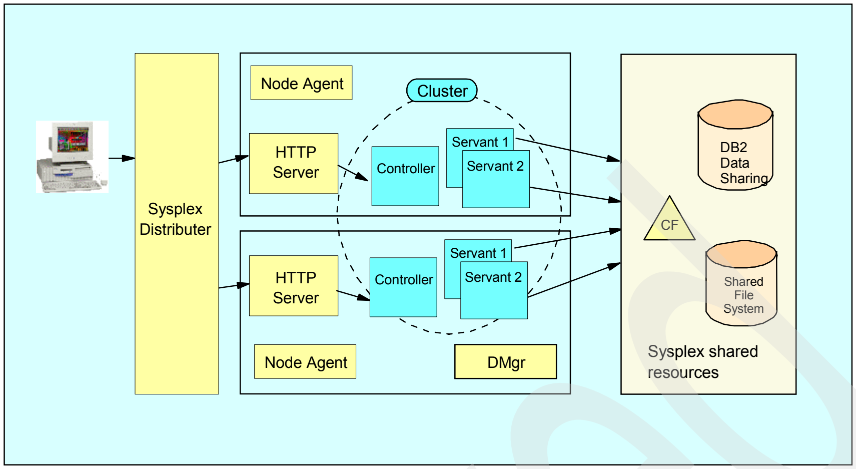


Figure 5-4 WebSphere/Portal system HA level 4

Single points of failure include:

- None
- Still vulnerable to disasters

5.3.6 WebSphere/Portal HA level 5

HA level 5 systems are used in disaster recovery scenarios where protection of an entire data center is required. This is the only system discussed this far that supports the concept of continuous availability. In this scenario, two identical configurations are created. When one site becomes unavailable, the other can take over either automatically or through a set of automated procedures. How fast the takeover happens is determined by several factors:

- A disaster recovery implementation implies geographical separation of the primary and secondary sites. How far apart the systems are is one factor in determining how fast the backup site can take over the workload. The further apart the systems are, the more expensive the hardware extenders are likely to be to support the communications links.
- How much of the data must be synchronized? How often is it replicated and/or synchronized? The shorter the replication interval, the less data will be needed to be synchronized at the time of a failure.
- If true continuous availability is required, the backup site must be a hot standby site in all respects. GDPS techniques such as dual write DASD, Peer to Peer Remote Copy (PPRC), Extended Remote Copy, and HyperSwap

need to be employed. In addition, high speed, high bandwidth networks must also be employed to enable the efficient transfer of data.

In an ideal HA level 5 deployment, the backup data center hosts an exact copy of the production environment. All configuration is identical with the exception of the active IP address used to access the site. In the event of a site failure, the only change required to transfer the load to the backup is a reconfiguration of the primary DNS address for the backup site.

In a situation where the primary and backup data centers are both actively sharing the production load and are within the same WebSphere administrative cell, the DNS update and associated routing commands will be handled automatically by the network infrastructure.

If the primary and backup sites are located in their own WebSphere cells, the DNS updates are manual. However, the commands to accomplish this can be automated to expedite the takeover.

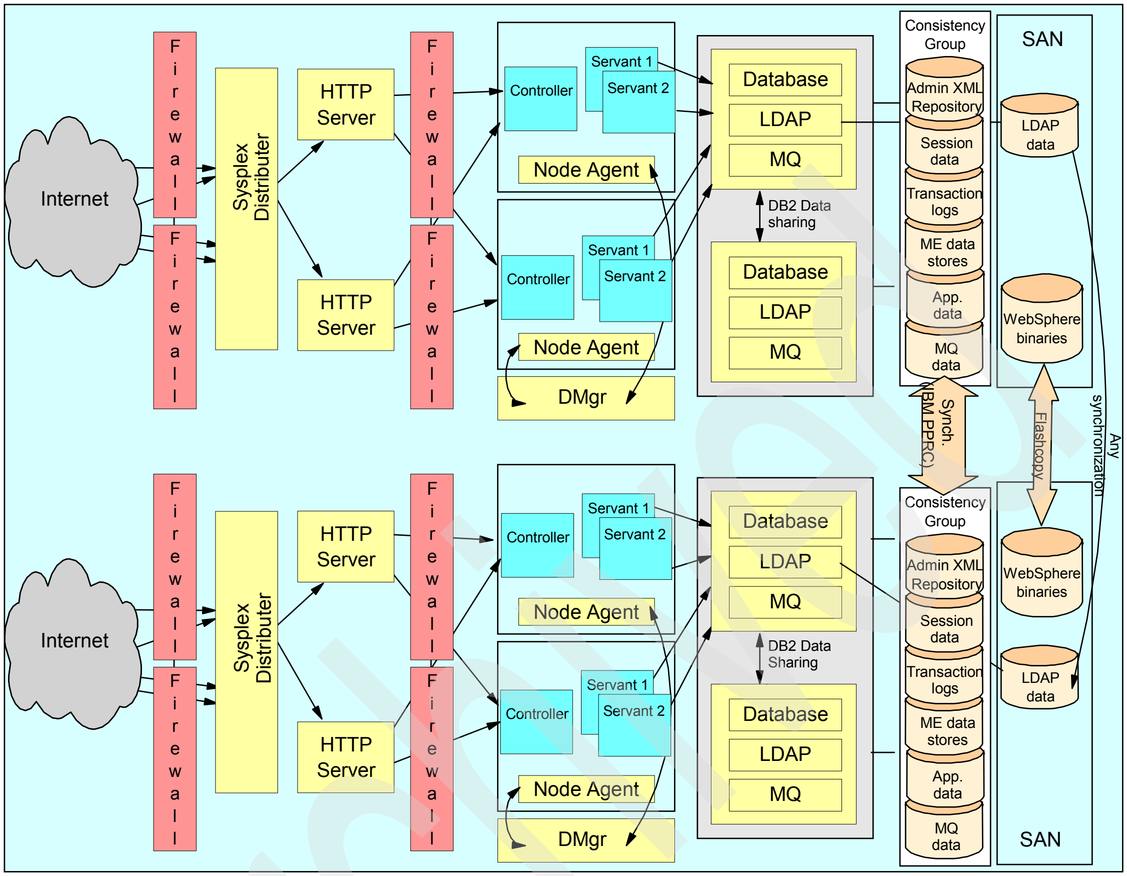


Figure 5-5 WebSphere/Portal system HA level 5

HA level 5 provides the following features:

- All firewalls have a backup.
- All load balancers have a backup.
- There are two or more HTTP servers.
- The combination of vertical scaling on an LPAR and horizontal scaling across the sysplex offers failover protection against operating system failures and hardware failures.
- DB2 data sharing in employed for all database resident data (configuration data, application data, session data, messaging data, directory data, and so on) providing full backup and recovery capabilities.
- The environment is protected across a disaster situation. Data is duplexed across a backup site using GDPS capabilities (PPRC, XRC, HyperSwap, and so on).

6

Implementing high availability with WebSphere Portal on z/OS

This chapter discusses the steps involved in creating a high availability environment with multiple WebSphere Portal application servers. Each step for creating the clustered application servers is reviewed in detail. We also describe some high availability considerations for other environmental components used by WebSphere Portal.

6.1 Considerations

To achieve high availability in the WebSphere Application Server arena, and for its add-ons like WebSphere Portal, WebSphere ESB (WebSphere Enterprise Service Bus), WebSphere Process Server (WPS), you need to implement and exploit a few techniques, some of which are traditional and used in many circumstances. We address those techniques in the sections which follow.

- WebSphere Portal uses DB2 databases. Those databases will need to be shared. Remember that 17 databases are currently used by WebSphere Portal.
- The HFS file system will need to be shared as well, although for performance reasons such attributes as LPAR ownership and access intent (R/O, R/W) have to be evaluated.
- Sysplex Distributor (SD) and the usage of Dynamic Virtual IP Addressing (DVIPA) also play an important role in the availability and load balancing of systems.

By putting clustering in place for the WebSphere Portal, we will exploit these features.

6.1.1 Our design

Our design was composed of two LPARs (SC49 and SC54). On each additional system we created a managed node for the WebSphere Application Server. Each node would contain a WebSphere Portal application clone. Remember that the Portal is an application server which has special programs installed. The current Portal in SC49 would be the seed for the cluster. Clones of this one would be built in the other LPARs, and together they would form the cluster.

Each node would have its own file system, with dedicated ownership and R/W authority. For availability reasons the J2EE programs (Portal programs and adjunct programs), when deployed to the cluster, would be physically stored in each of the filesystems.

For all Portal application servers belonging to the cluster we planned to use the same port numbers, so that some could be defined on the Sysplex Distributor. This was also the case for some ports of the daemon and the Node Agent. Keep in mind that the maximum number of ports that can be defined on the Sysplex Distributor is 64.

Building a cluster requires careful planning, so for guidance we recommend that you read white paper WP100653, “WebSphere for z/OS V6 Sample ND Configuration” which can be obtained from this site:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100653

It is also helpful to read about the clustering scenarios in the WebSphere Portal for z/OS Infocenter.

6.2 Building the cluster

Readers who have built clusters for a regular WebSphere Application Server know that this work is done partially via the ISPF dialog, but also partially via the WebSphere Administrative console.

For WebSphere Portal, all work is prepared with the ISPF customization dialogs and executed via batch jobs. The existing Portal application server is the primary in the primary nodes, and all others are secondaries in secondary nodes.

6.2.1 Overview - order of operations for clustering

It is important to note the order in which the tasks for clustering the WebSphere Portal application servers are performed. We begin this process with the assumption that we already have a WebSphere Portal application server on our primary node that uses a DB2 back-end database with “global security” enabled. (This was achieved after performing the tasks described in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35 through Chapter 4, “Securing WebSphere Portal with LDAP” on page 109.)

Consequently, the following work has to be executed:

1. Disable global security in the Deployment Manager.

   Refer to 6.2.2, “Disabling global security” on page 168.

2. Create an empty node on the second LPAR and federate it into the existing cell.

   Refer to 6.2.3, “Creating an empty WebSphere Application Server node on the second LPAR and federating” on page 168.

3. Install the WebSphere Portal base code into second node.

   Refer to 6.2.4, “Installing WebSphere Portal base code into second node” on page 186.

4. Build the cluster definition on the primary node.

Refer to 6.2.5, “Cluster Portal on the primary node” on page 199.

5. Make a clone of the primary WebSphere Portal application server and add it to cluster on the secondary node.

   Refer to 6.2.6, “Configuring the Portal cluster in the secondary node” on page 204.

6. Enable security on the primary and secondary node.

   Refer to 6.2.7, “Reenabling security on primary node and secondary node” on page 211.

6.2.2 Disabling global security

In our scenario, prior to performing any steps for clustering we temporarily disabled global security in the cell, because creating a cluster with security on is much more difficult.

To disable global security, in the WebSphere Administrative Console select **Security** -> **Global Security**. Deselect the check box under the general properties heading Enable Global Security. Save the changes that were made and recycle the Node Agent and Deployment Manager.

6.2.3 Creating an empty WebSphere Application Server node on the second LPAR and federating

A WebSphere Application Server node must be created on the second LPAR before you can continue with the Portal cluster. The secondary node is created via the WebSphere Application Server ISPF dialog.

Start the ISPF dialog on the second LPAR, as follows:

```
exec 'BBWP6054.SBBOCLIB(BBOWSTRT)' 'APPL(PS2) PROD(EJP) PRODHLQ(BBWP6054)'
```

> **Note:** It is very important that you specify a different APPL parameter from that used when you configured the primary node. Otherwise, your configuration will become confused with a mixture of variables from the primary and secondary nodes.
>
> In our case, we used APPL(PS2) to signify the variables associated with the secondary node.

After the dialog starts, select option **3**:

```
3  Create Network Deployment cells and nodes.  You must complete
   Option 1 before starting this option.
```

Next, choose option **2** to create the empty node:

```
2  Create an empty managed node and add it to an existing Network
   Deployment cell.  The managed node will contain a node agent
   but no application servers.  Create and manage application
   servers in the node using the administrative console or
   scripting.
```

Load the security domain variables used in previous tasks by choosing option 1:

```
1  Load security domain variables. Load your security domain
variables.
```

Load the customization variables used in previous tasks by choosing option L:

```
L  Load customization variables. Load your customization variables
from a data set.
```

> **Note:** You can save time and reduce the chance of typing errors by loading the variables saved when you configured the first empty node. Then you only need to change fields with the LPAR name and hostname, and change the suffix for various names from a 1 to 2 or an A to a B, depending on your naming convention.
>
> However, you must be sure that the variables you are loading contain *only* WebSphere variables and no Portal variables. Check this by looking for variables prefixed EJP in the SAVECFG you intend to load from. It is easy to inadvertently save variables to the wrong SAVEFG dataset and you must be sure that you do not load any primary node Portal variables in the secondary node.

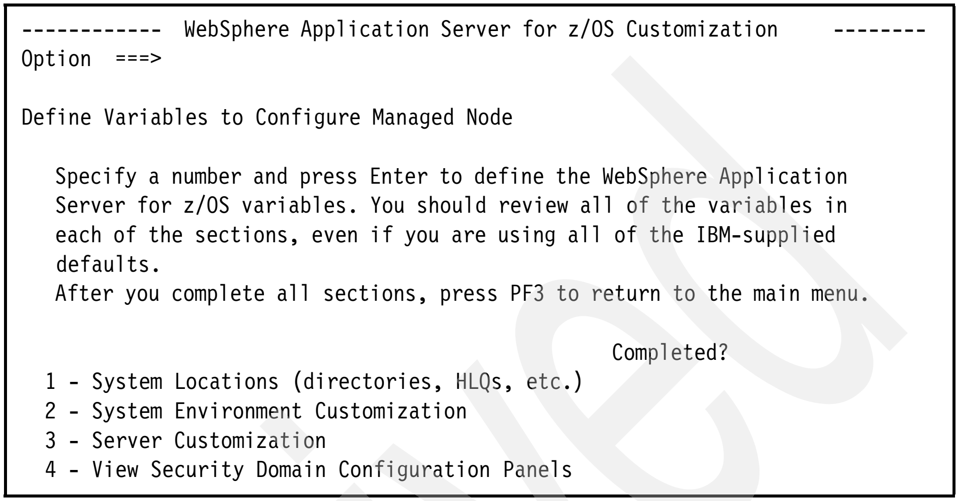
Define the datasets where the install jobs are stored by choosing option **2**:

```
2  Allocate target data sets. The data sets will contain the
   customization jobs and data generated by the dialog.
```

Now define the variables that will be used to generate the node creation jobs by choosing option **3**:

```
3  Define variables. Define your installation-specific information
for customization.
```

Start entering the system variables by choosing option **1 System Locations** in the Define Variables to Configure Managed Node panel, as shown in Figure 6-1. We eventually must go through each of these options.

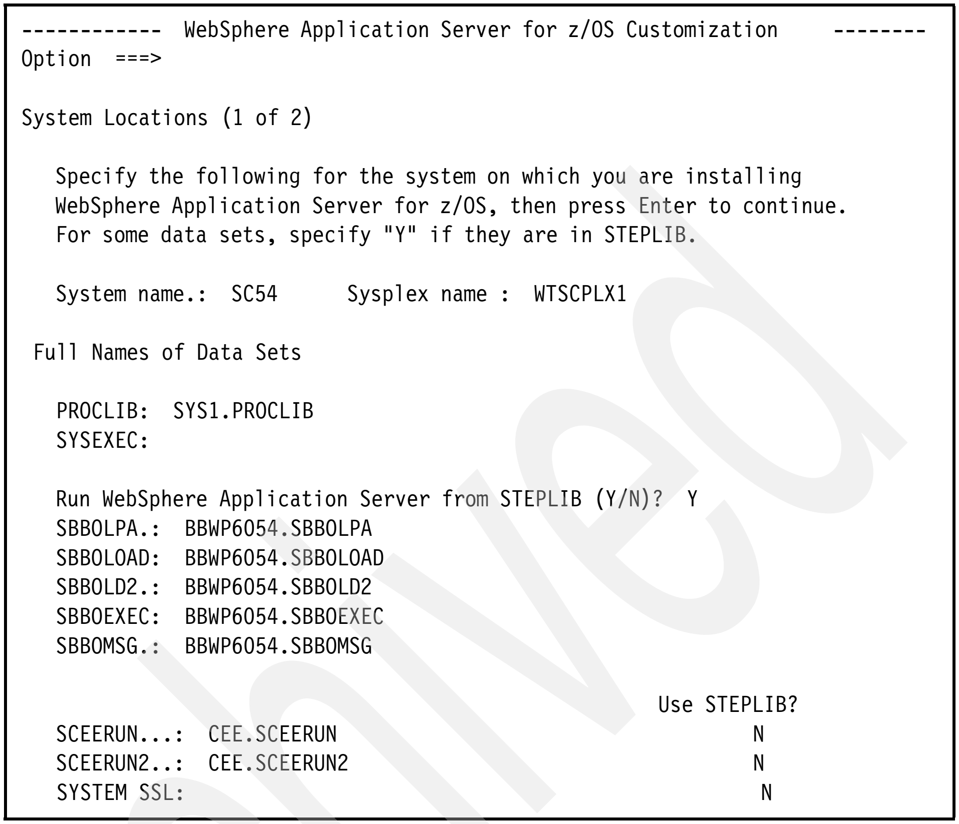


```
------------  WebSphere Application Server for z/OS Customization    --------
Option  ===>

Define Variables to Configure Managed Node

   Specify a number and press Enter to define the WebSphere Application
   Server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.

                                                    Completed?
  1 - System Locations (directories, HLQs, etc.)
  2 - System Environment Customization
  3 - Server Customization
  4 - View Security Domain Configuration Panels
```

Figure 6-1 Create an empty WebSphere Application Server node - system locations

The system variables used in the panel shown in Figure 6-2 on page 171 for the secondary node are very similar to those that were entered during the WebSphere Application Server install:

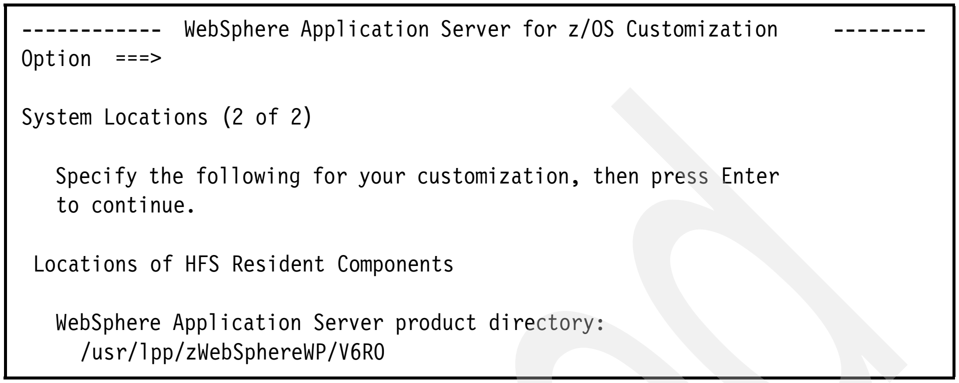
- `System name` is the name of the LPAR that the secondary node is created on.
- The `Sysplex name` should already be filled in if this is a sysplex. In our case, we do not have any of the required data sets in the LINKLST, so we run WebSphere Application Server code from STEPLIB.
- All of the SBBO* libraries require the fully qualified data set names used on the LPAR for the WebSphere Application Server data sets.
- The SCEERUN libraries are set to the fully qualified data set names of the language environment data sets.
- Our system does not use system SSL.



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

System Locations (1 of 2)

   Specify the following for the system on which you are installing
   WebSphere Application Server for z/OS, then press Enter to continue.
   For some data sets, specify "Y" if they are in STEPLIB.

   System name.:  SC54      Sysplex name :  WTSCPLX1

 Full Names of Data Sets

   PROCLIB:  SYS1.PROCLIB
   SYSEXEC:

   Run WebSphere Application Server from STEPLIB (Y/N)?  Y
   SBBOLPA.:  BBWP6054.SBBOLPA
   SBBOLOAD:  BBWP6054.SBBOLOAD
   SBBOLD2.:  BBWP6054.SBBOLD2
   SBBOEXEC:  BBWP6054.SBBOEXEC
   SBBOMSG.:  BBWP6054.SBBOMSG

                                                    Use STEPLIB?
   SCEERUN...:  CEE.SCEERUN                                N
   SCEERUN2..:  CEE.SCEERUN2                               N
   SYSTEM SSL:                                              N
```

Figure 6-2 Create an empty WebSphere Application Server node - System Locations (1 of 2)

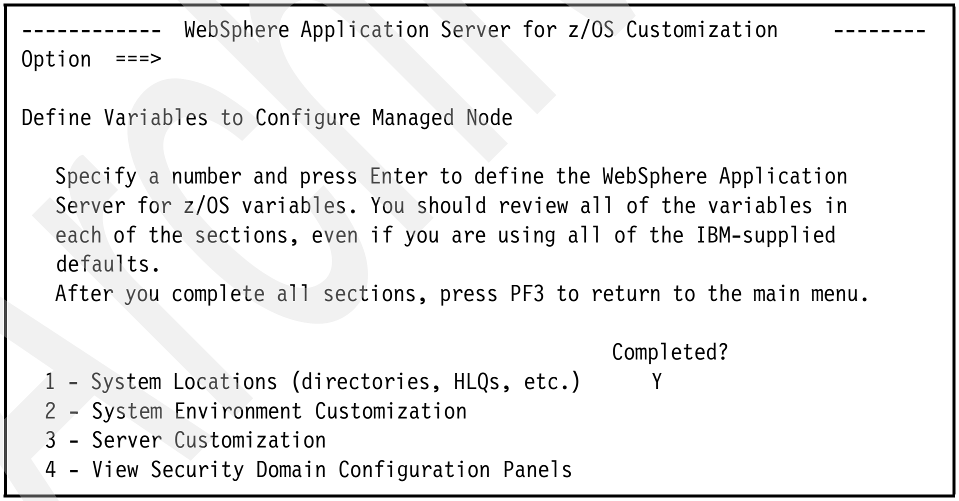
The only parameter that has to be entered on the panel shown in Figure 6-3 is the full directory path of the WebSphere Application Server code.



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

System Locations (2 of 2)

   Specify the following for your customization, then press Enter
   to continue.

 Locations of HFS Resident Components

   WebSphere Application Server product directory:
     /usr/lpp/zWebSphereWP/V6R0
```

Figure 6-3 Create an empty WebSphere Application Server node - System locations (2 of 2)

The System locations variables are now complete. Now, select option **2 System Environment Customization** in the Define Variables to Configure Managed Node panel, as shown in Figure 6-4.

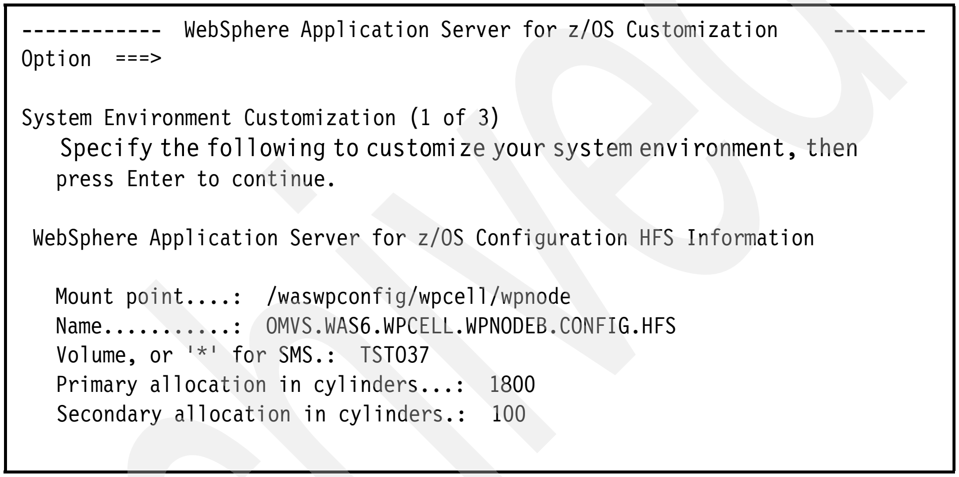


```
------------  WebSphere Application Server for z/OS Customization    --------
Option  ===>

Define Variables to Configure Managed Node

   Specify a number and press Enter to define the WebSphere Application
   Server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.

                                                   Completed?
  1 - System Locations (directories, HLQs, etc.)      Y
  2 - System Environment Customization
  3 - Server Customization
  4 - View Security Domain Configuration Panels
```

Figure 6-4 Create an empty WebSphere Application Server node - System Environment Customization

In the next panel, shown in Figure 6-5, the first variable here is the mount point of the HFS. The next variable is the actual data set that will be mounted at the mount point above it. If the HFS data set should be defined to a specific volume or volumes, enter them here or use an asterisk (*) if the HFS data set can be defined to SMS-managed volumes.

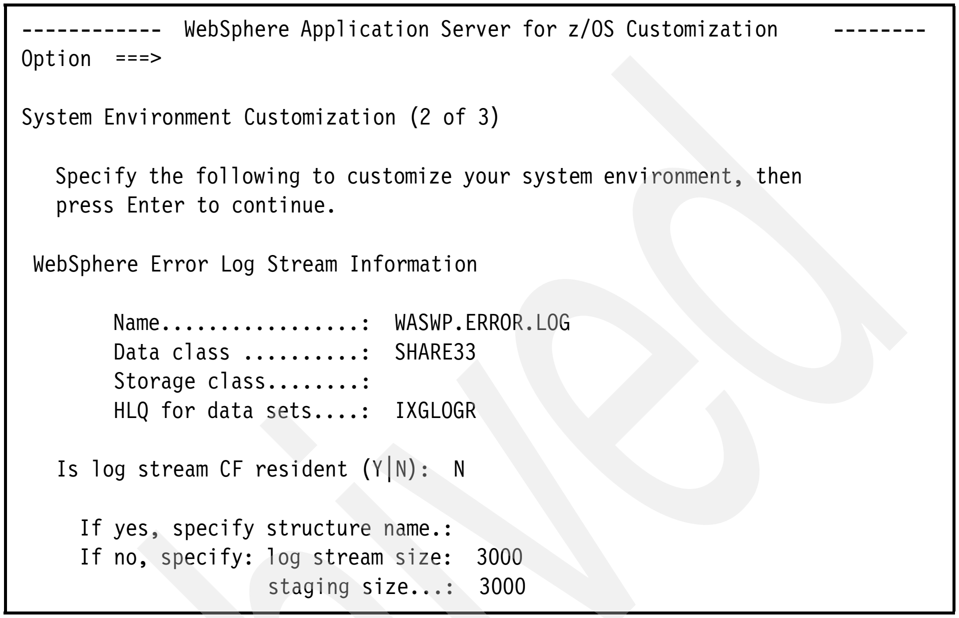
Primary and secondary allocation need to be at least as large as the minimum defined by WebSphere Portal. This is discussed in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35. WebSphere Portal requires a minimum of 1600 cylinders for a primary allocation, so the secondary node must also have at least 1600 cylinders allocated.



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

System Environment Customization (1 of 3)
   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Application Server for z/OS Configuration HFS Information

   Mount point....:  /waswpconfig/wpcell/wpnode
   Name...........:  OMVS.WAS6.WPCELL.WPNODEB.CONFIG.HFS
   Volume, or '*' for SMS.:  TST037
   Primary allocation in cylinders...:  1800
   Secondary allocation in cylinders.:  100
```

Figure 6-5 Create an empty WebSphere Application Server node - System Environment Customization (1 of 3)

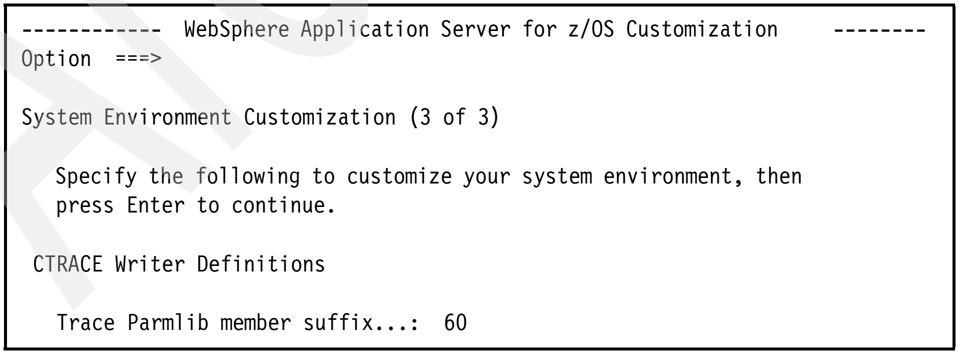
In our case, we did not set up logging on our system (see Figure 6-6). See the WebSphere Application Server V6 for z/OS information center for more information about the WebSphere error log stream.



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

System Environment Customization (2 of 3)

   Specify the following to customize your system environment, then
   press Enter to continue.

 WebSphere Error Log Stream Information

        Name.................:  WASWP.ERROR.LOG
        Data class ..........:  SHARE33
        Storage class........:
        HLQ for data sets....:  IXGLOGR

   Is log stream CF resident (Y|N):  N

     If yes, specify structure name.:
     If no, specify: log stream size:  3000
                     staging size...:  3000
```

Figure 6-6 Create an empty WebSphere Application Server node - System Environment Customization (2 of 3)

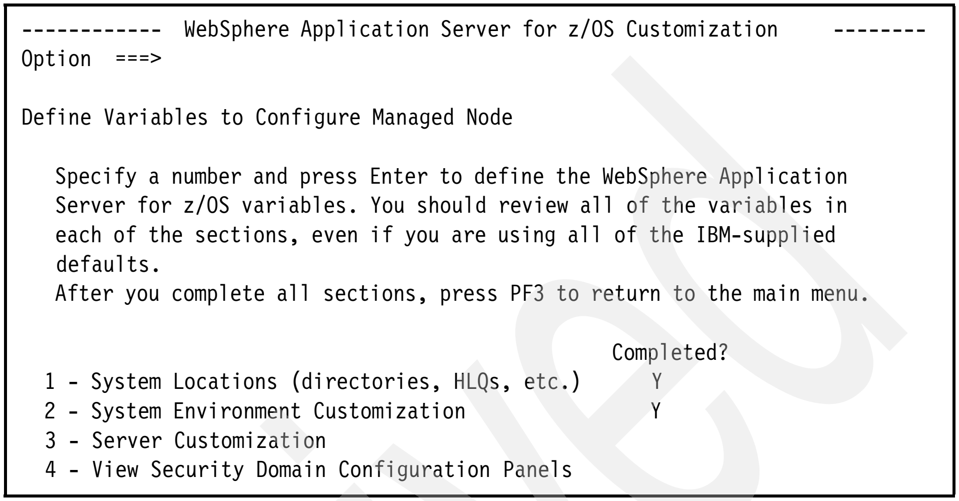
We did not set up the CTRACE writer on our system (see Figure 6-7). See the WebSphere Application Server V6 for z/OS information center for more information about the CTRACE writer.



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

System Environment Customization (3 of 3)

   Specify the following to customize your system environment, then
   press Enter to continue.

 CTRACE Writer Definitions

   Trace Parmlib member suffix...:  60
```

Figure 6-7 Create an empty WebSphere Application Server node - System Environment Customization (3 of 3)

Sections 1 and 2 have now been completed. Select option **3 Server Customization** in the Define Variables to Configure Managed Node panel, as shown in Figure 6-8.



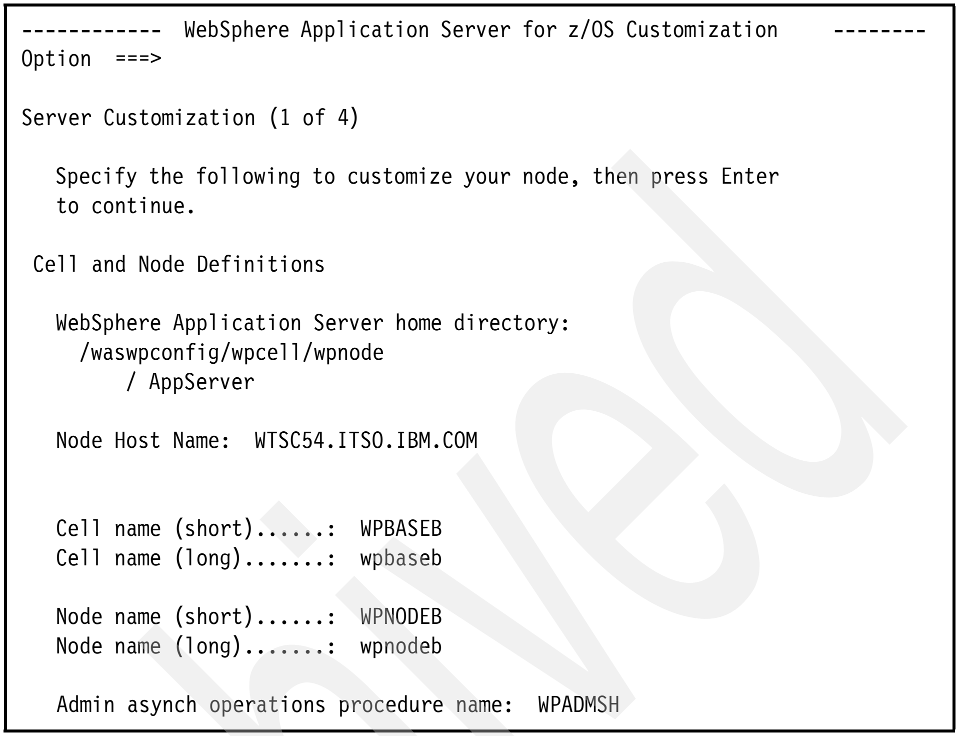
```
-----------  WebSphere Application Server for z/OS Customization    --------
Option  ===>

Define Variables to Configure Managed Node

   Specify a number and press Enter to define the WebSphere Application
   Server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.

                                                   Completed?
  1 - System Locations (directories, HLQs, etc.)      Y
  2 - System Environment Customization                Y
  3 - Server Customization
  4 - View Security Domain Configuration Panels
```

Figure 6-8 Create an empty WebSphere Application Server node - Server Customization

In the next panel, as shown in Figure 6-9 on page 176, the WebSphere Application Server home directory is filled in with the mount point that was entered into the first customization panel. We chose the default for the subdirectory of the mount point, which is AppServer.

We explain the other variables here:

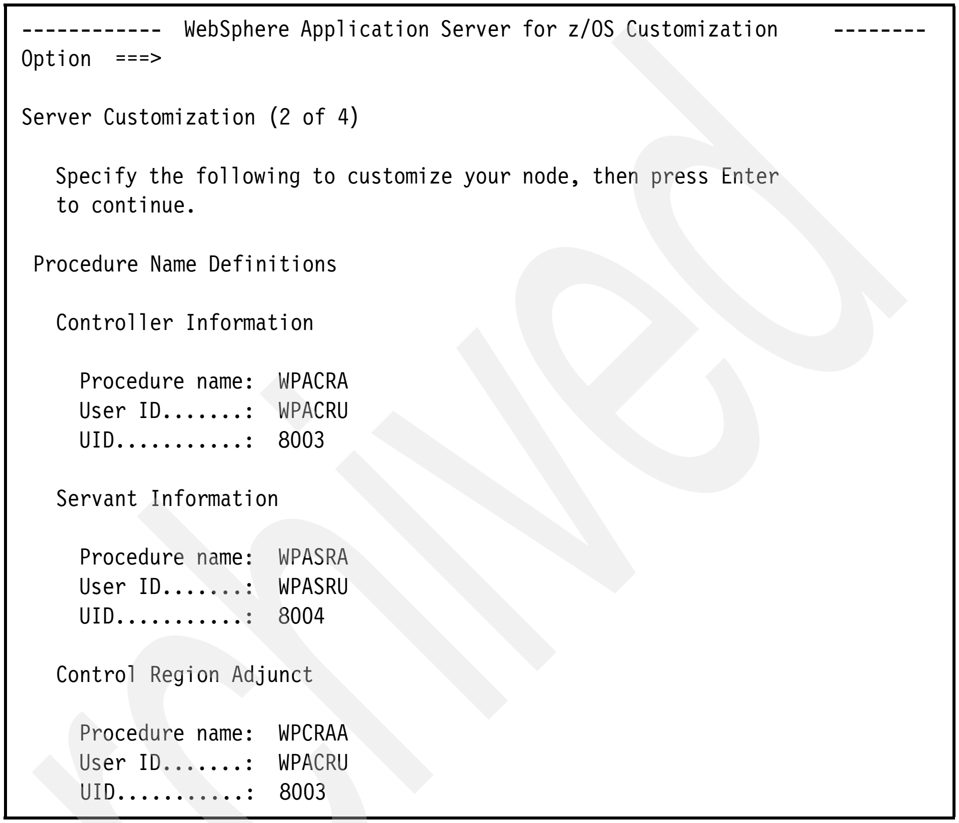
- `Node Host Name` is the name of the LPAR that the node is being created on. Do not use the DVIPA name if there is a Sysplex Distributor being utilized.
- `Cell name (short)` and `Cell name (long)` are only temporary names. A cell will be created on the secondary LPAR when the node is initially defined with these names. After the cell and node are defined, the secondary node will be federated into the primary managed node and the cell that was defined here will be removed.
- `Node name (short)` and `Node name (long)` are the permanent node names, so make sure the names are significant to the environment naming conventions.
- `Admin asynch operations procedure name` specifies the JCL procedure name of the started task that is launched by Node Agents or application servers to perform certain asynchronous administrative operations such as node synchronization.



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

Server Customization (1 of 4)

   Specify the following to customize your node, then press Enter
   to continue.

 Cell and Node Definitions

   WebSphere Application Server home directory:
     /waswpconfig/wpcell/wpnode
         / AppServer

   Node Host Name:  WTSC54.ITSO.IBM.COM


   Cell name (short)......:  WPBASEB
   Cell name (long).......:  wpbaseb

   Node name (short)......:  WPNODEB
   Node name (long).......:  wpnodeb

   Admin asynch operations procedure name:  WPADMSH
```

Figure 6-9 Create an empty WebSphere Application Server node - Server Customization (1 of 4)

The next panel, shown in Figure 6-10, lists all of the IDs that are used by the WebSphere Application Server address spaces and the procedures that they are associated with. These IDs must be predefined in RACF and in USS before the panel can be filled out.

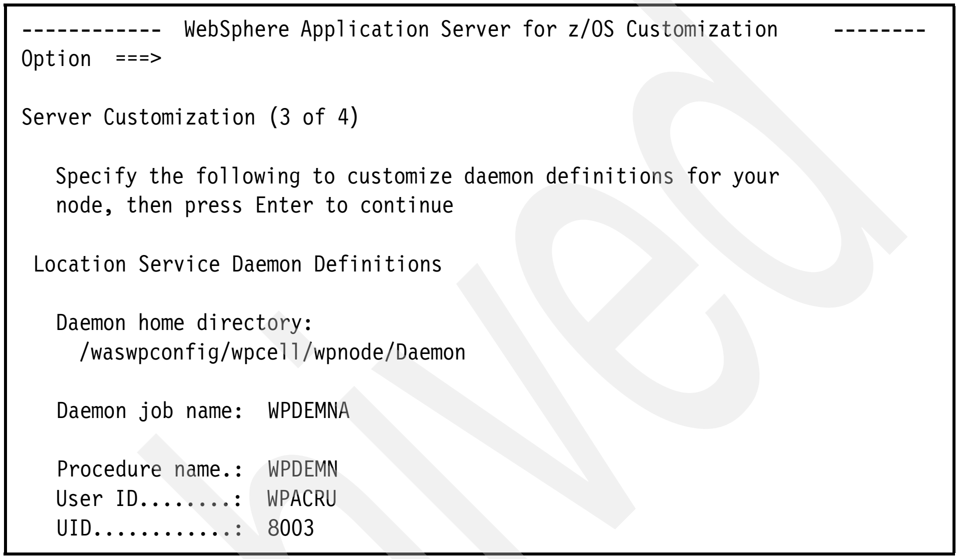


```
------------  WebSphere Application Server for z/OS Customization    --------
Option  ===>

Server Customization (2 of 4)

   Specify the following to customize your node, then press Enter
   to continue.

 Procedure Name Definitions

   Controller Information

     Procedure name:  WPACRA
     User ID.......:  WPACRU
     UID...........:  8003

   Servant Information

     Procedure name:  WPASRA
     User ID.......:  WPASRU
     UID...........:  8004

   Control Region Adjunct

     Procedure name:  WPCRAA
     User ID.......:  WPACRU
     UID...........:  8003
```

Figure 6-10 Create an empty WebSphere Application Server node - Server Customization (2 of 4)

The next panel, shown in Figure 6-11, asks for information pertaining to the WebSphere Application Server daemon. We explain the variables here:

- `Daemon job name` is the name that will be used for the daemon started task.
- `Procedure name` is the name of the JCL member is SYS1.PROCLIB.
- `User ID` is the ID that will run the daemon started task.
- `UID` is the USS UID of the RACF ID running the daemon started task.



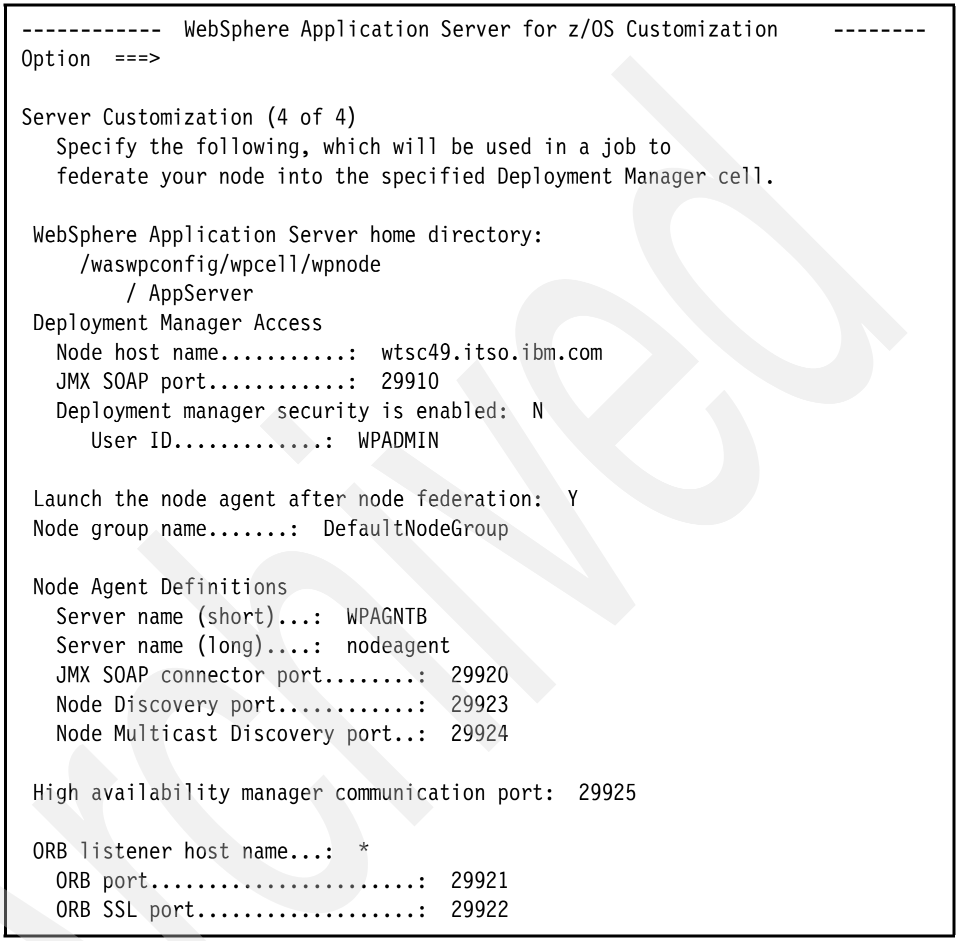
```
------------  WebSphere Application Server for z/OS Customization     --------
Option  ===>

Server Customization (3 of 4)

   Specify the following to customize daemon definitions for your
   node, then press Enter to continue

 Location Service Daemon Definitions

   Daemon home directory:
     /waswpconfig/wpcell/wpnode/Daemon

   Daemon job name:  WPDEMNA

   Procedure name.:  WPDEMN
   User ID........:  WPACRU
   UID............:  8003
```

Figure 6-11 Create an empty WebSphere Application Server node - Server Customization (3 of 4)

The last server customization panel, shown in Figure 6-12 on page 179, asks for information about the Deployment Manager. We explain the variables here:

- `Node host name` is the host name of the LPAR that the Deployment Manager is running on.
- `JMX SOAP port` is the SOAP port of the Deployment Manager.
- As mentioned in 6.2.1, security should be disabled before starting to build the cluster. For this reason, set `Deployment manager security is enabled` to `N`.
- In our case, we entered the Deployment Manager administrator ID even though security is disabled. However, it can be left blank.
- Because we are building a cluster, we want to launch the nodeagent after we federate the node. In our case, we took the default node group name on our primary and secondary nodes.
- `Server name (short)` is the name of the Node Agent address space. `Server name (long)` is the name that is displayed in the administrative console for the Node Agent under **System Administration** -> **Node agents**.

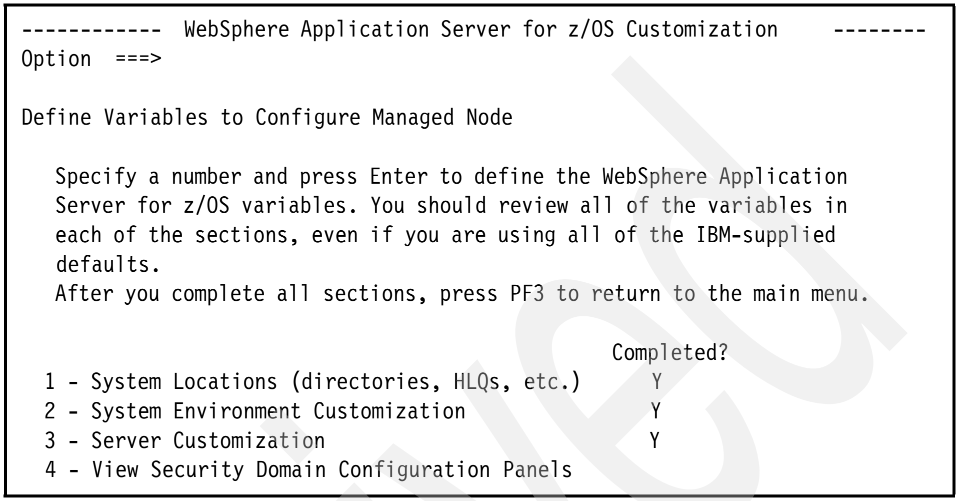
The remaining parameters are the ports defined on the secondary LPAR. Make sure they are unique to the LPAR. They can be the same ports that were used on the primary LPAR.



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

Server Customization (4 of 4)
   Specify the following, which will be used in a job to
   federate your node into the specified Deployment Manager cell.

 WebSphere Application Server home directory:
     /waswpconfig/wpcell/wpnode
         / AppServer
 Deployment Manager Access
   Node host name...........:  wtsc49.itso.ibm.com
   JMX SOAP port............:  29910
   Deployment manager security is enabled:  N
      User ID.............:  WPADMIN

 Launch the node agent after node federation:  Y
 Node group name.......:  DefaultNodeGroup

 Node Agent Definitions
   Server name (short)...:  WPAGNTB
   Server name (long)....:  nodeagent
   JMX SOAP connector port........:  29920
   Node Discovery port............:  29923
   Node Multicast Discovery port..:  29924

 High availability manager communication port:  29925

 ORB listener host name...:  *
   ORB port......................:  29921
   ORB SSL port..................:  29922
```

Figure 6-12 Create an empty WebSphere Application Server node - Server Customization (4 of 4)

Sections 1,2 and 3 of the panel Define Variables to Configure Managed Node are now complete. Select option **4 View Security Domain Configuration Panels**.

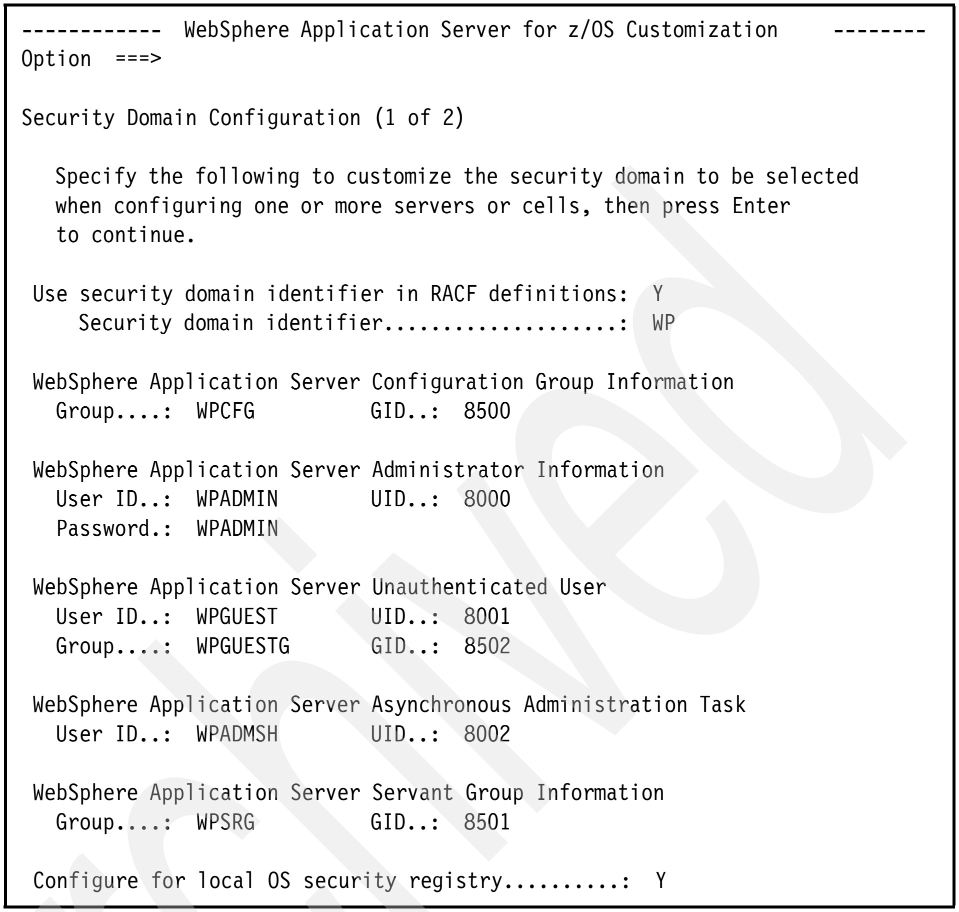


```
------------  WebSphere Application Server for z/OS Customization    --------
Option  ===>

Define Variables to Configure Managed Node

   Specify a number and press Enter to define the WebSphere Application
   Server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.

                                                   Completed?
  1 - System Locations (directories, HLQs, etc.)      Y
  2 - System Environment Customization                Y
  3 - Server Customization                            Y
  4 - View Security Domain Configuration Panels
```

Figure 6-13 Create an empty WebSphere Application Server node - View Security Domain

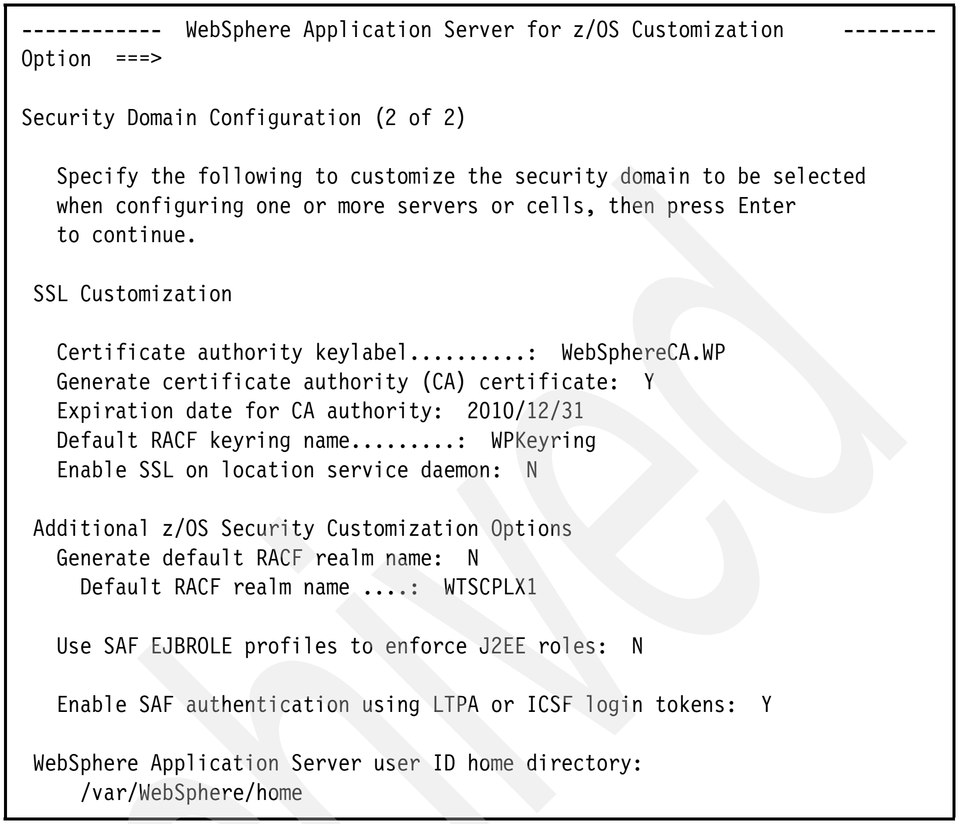
In the panels shown in Figure 6-14 on page 181 and Figure 6-15 on page 182, we did not alter the security variables because we installed our secondary node with security off.

Security is enabled on the Deployment Manager after we federate the node, so altering the two security domain panels is unnecessary.



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

Security Domain Configuration (1 of 2)

   Specify the following to customize the security domain to be selected
   when configuring one or more servers or cells, then press Enter
   to continue.

 Use security domain identifier in RACF definitions:  Y
     Security domain identifier....................:  WP

 WebSphere Application Server Configuration Group Information
   Group....:  WPCFG         GID..:  8500

 WebSphere Application Server Administrator Information
   User ID..:  WPADMIN       UID..:  8000
   Password.:  WPADMIN

 WebSphere Application Server Unauthenticated User
   User ID..:  WPGUEST       UID..:  8001
   Group....:  WPGUESTG      GID..:  8502

 WebSphere Application Server Asynchronous Administration Task
   User ID..:  WPADMSH       UID..:  8002

 WebSphere Application Server Servant Group Information
   Group....:  WPSRG         GID..:  8501

 Configure for local OS security registry..........:  Y
```

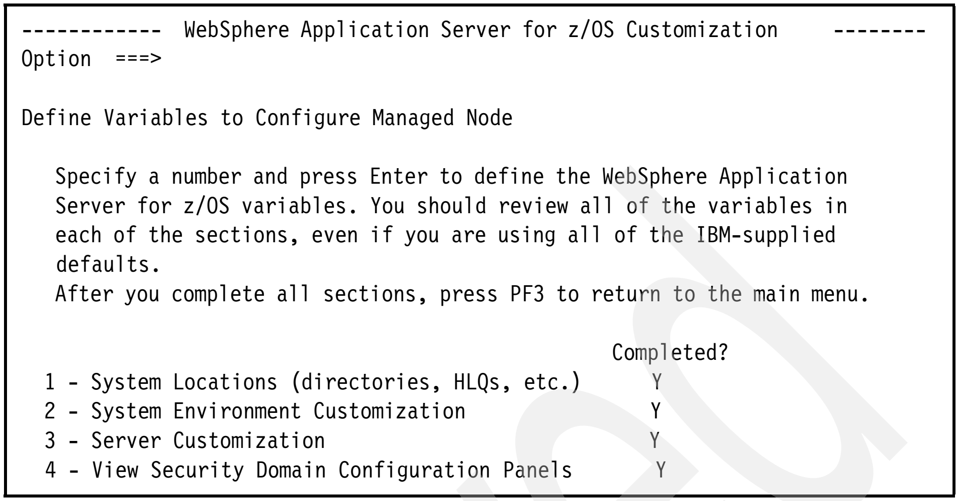
Figure 6-14 Create an empty WebSphere Application Server node - View Security Domain Configuration (1 of 2)



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

Security Domain Configuration (2 of 2)

   Specify the following to customize the security domain to be selected
   when configuring one or more servers or cells, then press Enter
   to continue.

 SSL Customization

   Certificate authority keylabel..........:  WebSphereCA.WP
   Generate certificate authority (CA) certificate:  Y
   Expiration date for CA authority:  2010/12/31
   Default RACF keyring name.........:  WPKeyring
   Enable SSL on location service daemon:  N

 Additional z/OS Security Customization Options
   Generate default RACF realm name:  N
     Default RACF realm name ....:  WTSCPLX1

   Use SAF EJBROLE profiles to enforce J2EE roles:  N

   Enable SAF authentication using LTPA or ICSF login tokens:  Y

 WebSphere Application Server user ID home directory:
     /var/WebSphere/home
```

Figure 6-15   Create an empty WebSphere Application Server node - View Security Domain Customization (2 of 2)

All the variables are now displayed as completed; see Figure 6-16 on page 183.



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

Define Variables to Configure Managed Node

   Specify a number and press Enter to define the WebSphere Application
   Server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.

                                                Completed?
  1 - System Locations (directories, HLQs, etc.)     Y
  2 - System Environment Customization               Y
  3 - Server Customization                           Y
  4 - View Security Domain Configuration Panels      Y
```

Figure 6-16 Completed node variables

Press F3 (Figure 6-16) to return to the main panel Configure Managed Node.

In the main panel, select option **4** to generate the jobs for this task.

```
4  Generate customization jobs. Validate your customization
variables and generate jobs and instructions.
```

After the jobs are generated, you can view the instructions for running the jobs by selecting option **5**, or by looking in the CNTL dataset at member BBOMNINS.

Before running any of the jobs, make sure that the following WebSphere Application Server data sets are APF-authorized.

```
BBWP6054.SBBOLPA
BBWP6054.SBBOLOAD
BBWP6054.SBBOLD2
```

Jobs in our installation scenario

Here we list the jobs that we addressed during our installation scenario. (There are some optional steps that did not apply to our system.) Note that we did not need RACF jobs, because we used the same IDs from the primary LPAR with a shared RACF. They are required to run, however, if you do not have shared RACF.

BBOMBRAJ

This job creates the RACF statements necessary to define the user IDs that are used to run the WebSphere Application Server address spaces.

We did not run this job because we had a RACF environment which was shared between our primary and secondary LPARs. We used the same IDs to run our address spaces on each LPAR, therefore no new RACF IDs were required.

BBOMBRAK

This job runs the RACF statements created by BBOMBRAJ.

We did not run this job for the same reason we did not run BBOMBRAJ.

BBOWMNMT

This job is used to create the mount point directory where all of the WebSphere Application Server node information is stored. It also creates the HFS data set that is mounted at the mount point directory. The user ID that runs this job must have a user ID of 0, because it creates a new HFS structure.

We changed the mount point directory definition after the job created it to use a symbolic link instead of the normal directory that is created by BBOWMNMT. This was done so that we could use the same USS path on both of the LPARs in our cluster. The symbolic link was created using the $SYSNAME variable to point the link to the property directory on each LPAR.

The following diagram, which is also included in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35, shows how the symbolic links were organized. On our secondary LPAR, we did not use the wpdmnode symbolic link on the secondary LPAR because the Deployment Manager node only existed on our primary LPAR.

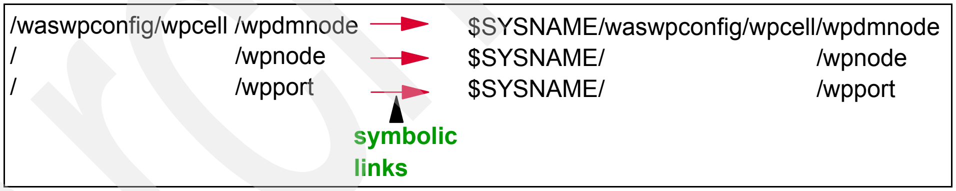


Figure 6-17 HFS organization with symbolic links

BBOMHFSA

This job populates the HFS created in BBOWMNMT. The user ID running BBOMHFSA must have a UID of 0 in USS.

BBOWCPYM

BBOWCPYM copies the customized started task procedures from the CNTL dataset where the WebSphere Application Server jobs were generated over to SYS1.PROCLIB.

We did not run this job because we had shared DASD between our LPARs. SYS1.PROCLIB is the same dataset on both LPARs, so we just used the existing procedures that were created when the primary node was built. If the environment does not have shared DASD, this job must be run.

BBOWWPFM
This job sets up the runtime HFS by creating the default profile for the node. If this job fails, the default profile directory must be deleted before the job can be rerun. The directory name that must be deleted is called WAS_HOME/profiles/default.

BBOWHFSB
BBOWHFSB completes the HFS installation and must run with a user ID that has a UID of 0.

BBOWMNAN
Before this job can be run, you must ensure that the Deployment Manager is active on the primary LPAR. BBOWMNAN federates the new, empty node on the secondary LPAR into the cell that was created on the primary LPAR. The job must be run with the WebSphere Application Server administrative ID which is WPADMIN on your primary system.

The ID WebSphere Application Server administrative ID must be defined to RACF on the secondary node, as well as the primary node. This ID must also have a USS home directory on the secondary LPAR.

After the BBOWMNAN job was run to federate the secondary node into the Deployment Manager, we were ready to clone the app server. Figure 6-17 on page 184 shows the configuration of the environment before WebSphere Portal is installed into the secondary node. Notice that LPAR wtsc54 only has the empty node, with no application servers defined.

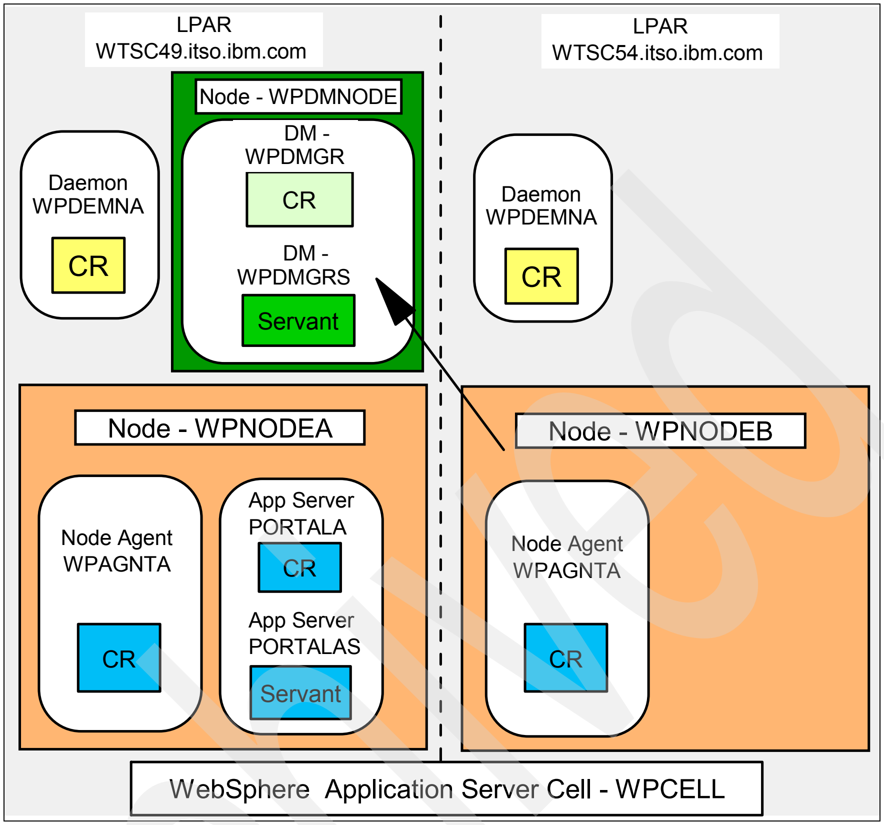


Figure 6-18 Network Deployment diagram with two nodes and one Portal application server

6.2.4 Installing WebSphere Portal base code into second node

In our case, now that we had an empty node federated into our Deployment Manager, we had to install the WebSphere Portal base code into the second node.

The panels to install the WebSphere Portal base code into the second node are similar to installing it into the primary node. The main difference is that fewer jobs are required because a portion of the Portal customization is obtained from the primary node when the cluster is built.

Entering WebSphere Portal variables into ISPF dialog

Start the WebSphere Application Server ISPF dialog, as follows:

```
exec 'BBWP6054.SBBOCLIB(BBOWSTRT)' 'APPL(PS2) PROD(EJP)
PRODHLQ(BBWP6054)'
```

Choose option **5**:

```
5  WebSphere Application Server-based add-on products. Configure
   other products that are built on WebSphere Application Server.
```

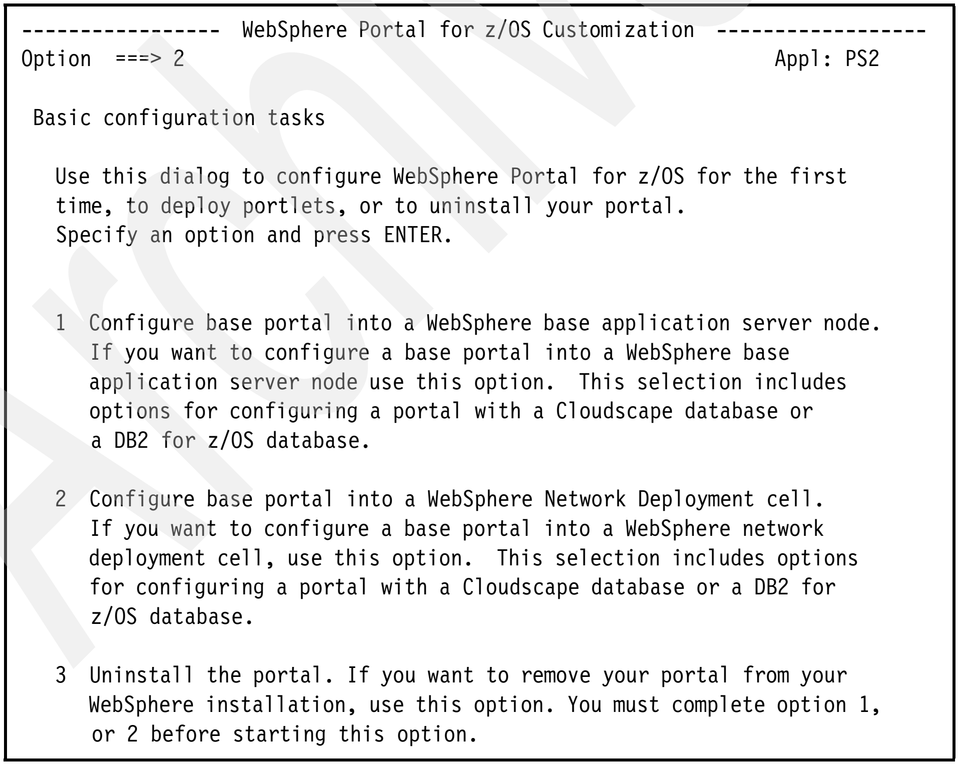
Select WebSphere Portal for z/OS:

```
1   WebSphere Portal for z/OS
    Configure WebSphere Portal
```

Press Enter to bypass the version screen. Choose option **1 - Basic configuration tasks**:

```
1  Basic configuration tasks. If you want to configure
   a base portal, use this option.
```

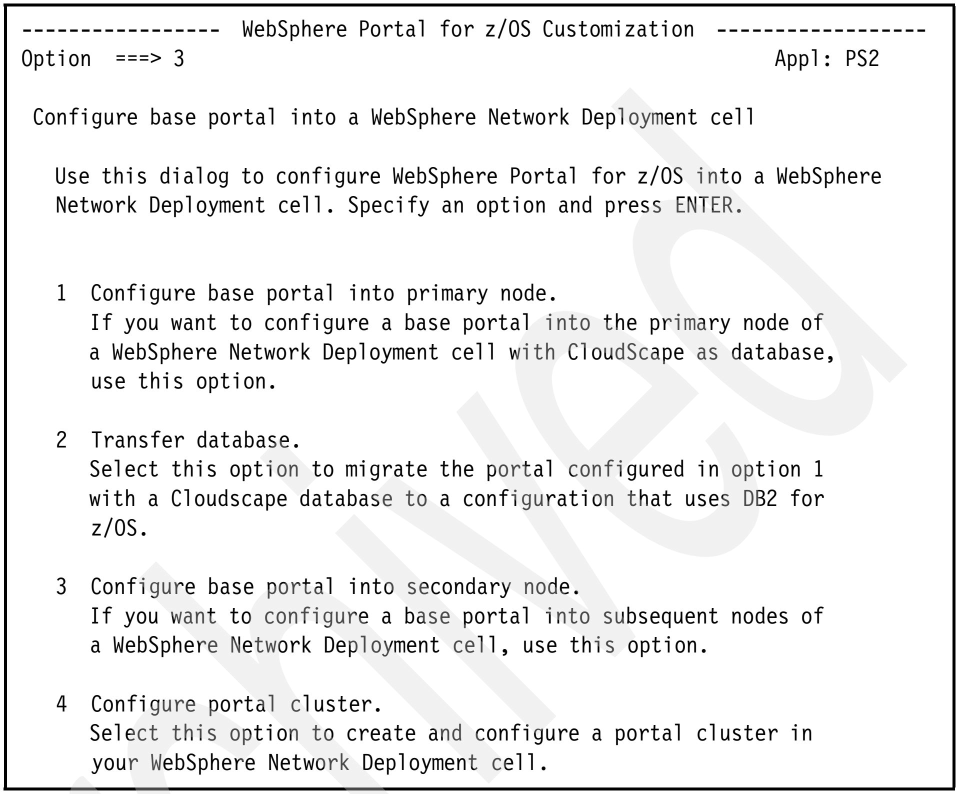
Enter option `2 - Configure base portal` into a WebSphere Network Deployment cell in the Basic configuration tasks panel, as shown in Figure 6-19.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 2                                                    Appl: PS2

 Basic configuration tasks

   Use this dialog to configure WebSphere Portal for z/OS for the first
   time, to deploy portlets, or to uninstall your portal.
   Specify an option and press ENTER.


   1  Configure base portal into a WebSphere base application server node.
      If you want to configure a base portal into a WebSphere base
      application server node use this option.  This selection includes
      options for configuring a portal with a Cloudscape database or
      a DB2 for z/OS database.

   2  Configure base portal into a WebSphere Network Deployment cell.
      If you want to configure a base portal into a WebSphere network
      deployment cell, use this option.  This selection includes options
      for configuring a portal with a Cloudscape database or a DB2 for
      z/OS database.

   3  Uninstall the portal. If you want to remove your portal from your
      WebSphere installation, use this option. You must complete option 1,
      or 2 before starting this option.
```

Figure 6-19 Installing WebSphere Portal into secondary node - Basic configuration tasks

To install the base WebSphere Portal code into a secondary node, select option 3 in the panel shown in Figure 6-20.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 3                                                  Appl: PS2

 Configure base portal into a WebSphere Network Deployment cell

   Use this dialog to configure WebSphere Portal for z/OS into a WebSphere
   Network Deployment cell. Specify an option and press ENTER.


   1  Configure base portal into primary node.
      If you want to configure a base portal into the primary node of
      a WebSphere Network Deployment cell with CloudScape as database,
      use this option.

   2  Transfer database.
      Select this option to migrate the portal configured in option 1
      with a Cloudscape database to a configuration that uses DB2 for
      z/OS.

   3  Configure base portal into secondary node.
      If you want to configure a base portal into subsequent nodes of
      a WebSphere Network Deployment cell, use this option.

   4  Configure portal cluster.
      Select this option to create and configure a portal cluster in
      your WebSphere Network Deployment cell.
```

Figure 6-20 Installing WebSphere Portal into secondary node - Configuring base portal into secondary node

Load the custom variables that you saved when configuring the *secondary* empty node using option `L`.

```
L  Load customization variables. Load your WebSphere Portal
   customization variables from a data set.
```

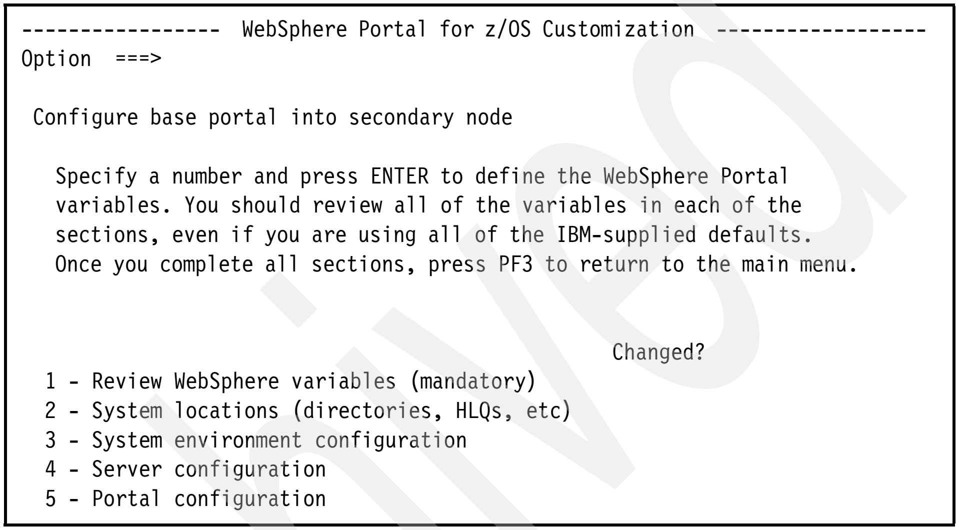
Select option **1** to verify the dataset where the install jobs will get generated. If no changes are made, the jobs generate into the same dataset that was used for all the other installation tasks.

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

Choose option **2** to enter the variables that are required for installing the base WebSphere Portal code into the secondary node.

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

The next panel, shown in Figure 6-21, displays all the categories of variables that need to be reviewed. Start by selecting option `1 - Review WebSphere variables`.



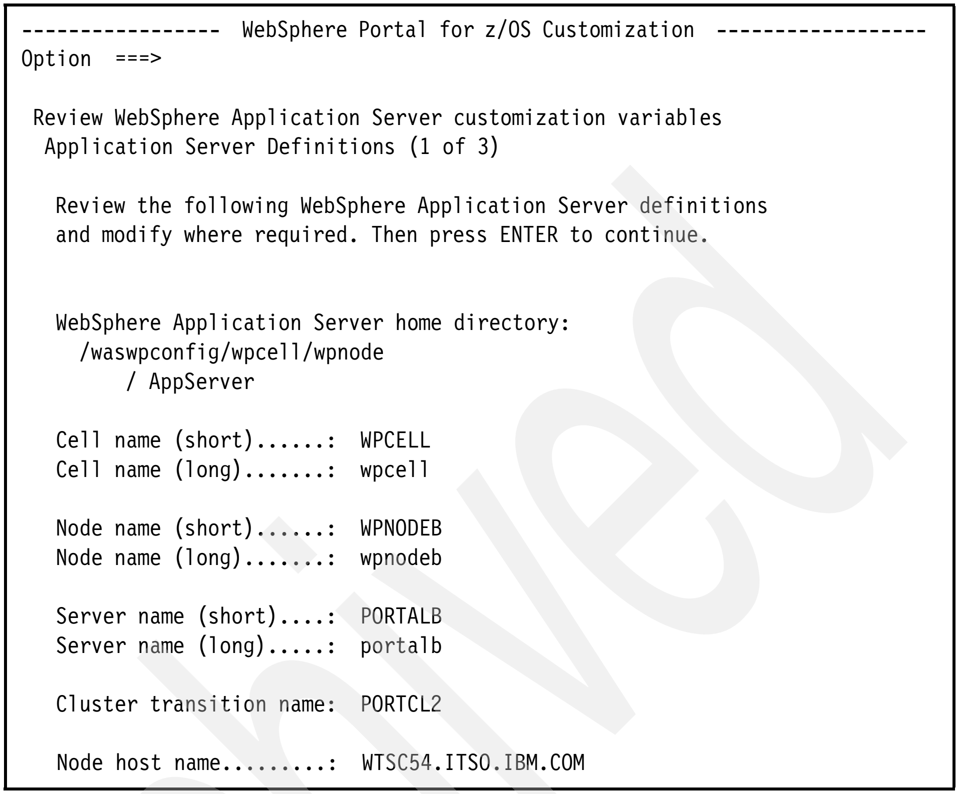
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===>

 Configure base portal into secondary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)
  2 - System locations (directories, HLQs, etc)
  3 - System environment configuration
  4 - Server configuration
  5 - Portal configuration
```

Figure 6-21   Installing WebSphere Portal into secondary node - Configure base portal into secondary node - reviewing variables

The next panel, shown in Figure 6-22 on page 190, is identical to Figure 2-16 on page 53. The difference is that the values are changed to represent the WebSphere Application Server node on the secondary LPAR.

The cell name is the same because the secondary node was federated into the cell on the primary node. The node name is the name of empty node that was created on the secondary LPAR. The server name and cluster transition name parameters are the names that, in our case, we will use for our new WebSphere Portal started task on the secondary node. The names entered here are copied into panel 2 of 3.

Make sure to use the names that are significant to the secondary LPAR because they cannot be altered in panel 2 of 3. `Node host name` is the name of the LPAR where the secondary node resides. Do not use DVIPA name if a Sysplex Distributor is being used. Using the DVIPA name causes communication errors with the secondary node being federated into the Deployment Manager.



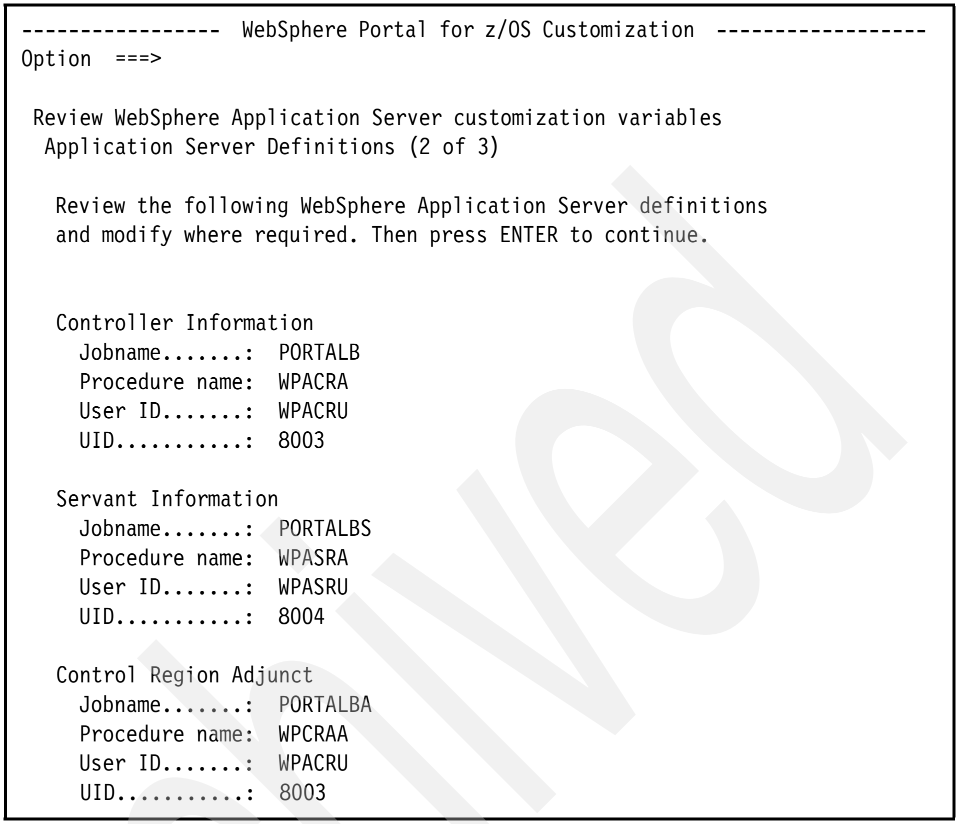
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Review WebSphere Application Server customization variables
  Application Server Definitions (1 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


   WebSphere Application Server home directory:
     /waswpconfig/wpcell/wpnode
         / AppServer

   Cell name (short)......:  WPCELL
   Cell name (long).......:  wpcell

   Node name (short)......:  WPNODEB
   Node name (long).......:  wpnodeb

   Server name (short)....:  PORTALB
   Server name (long).....:  portalb

   Cluster transition name:  PORTCL2

   Node host name.........:  WTSC54.ITSO.IBM.COM
```

Figure 6-22 Installing WebSphere Portal into secondary node - Configuring base portal into secondary node (1 of 3)

Panel 2 of 3 (Figure 6-23 on page 191) shows all the jobnames filled in. These names cannot be changed without going back to panel 1 of 3.

`Procedure name` is the name of the started task JCL procedure that is in SYS1.PROCLIB. WebSphere Portal uses the same procedures as WebSphere Application Server, so they already exist.

The user IDs on this panel must be defined in RACF before executing the batch jobs. They must also have an OMVS UID with an OMVS home directory. Failure to do this will cause many of the batch jobs to fail, because they write output to OMVS. In our case, we used the same IDs to run WebSphere Portal on the secondary node as we do on the primary node because we had a shared RACF database.

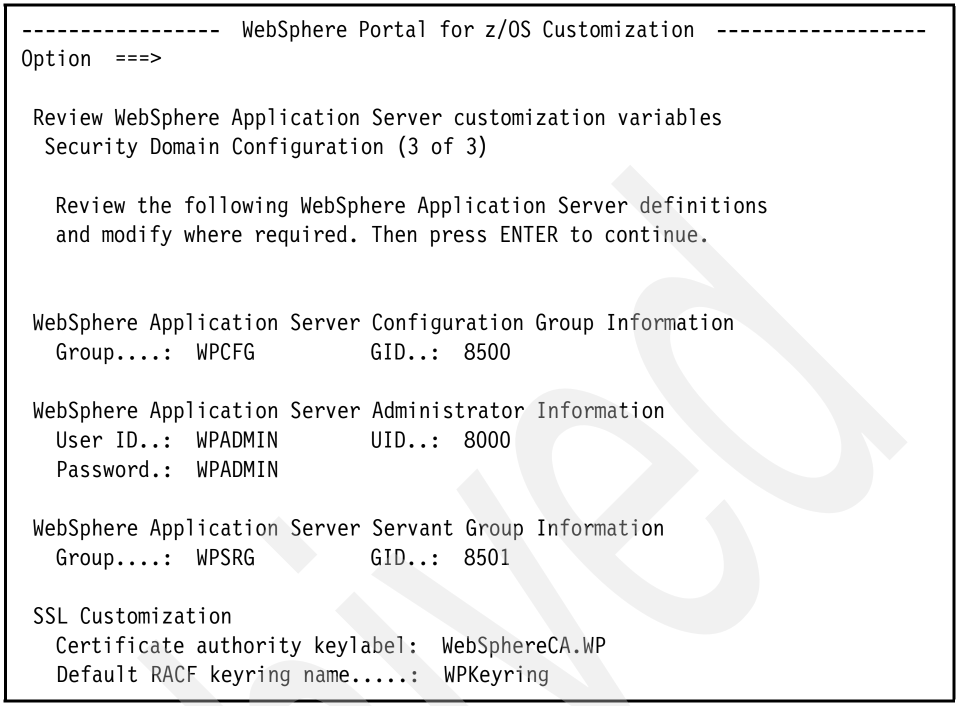


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Review WebSphere Application Server customization variables
  Application Server Definitions (2 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


   Controller Information
     Jobname.......:  PORTALB
     Procedure name:  WPACRA
     User ID.......:  WPACRU
     UID...........:  8003

   Servant Information
     Jobname.......:  PORTALBS
     Procedure name:  WPASRA
     User ID.......:  WPASRU
     UID...........:  8004

   Control Region Adjunct
     Jobname.......:  PORTALBA
     Procedure name:  WPCRAA
     User ID.......:  WPACRU
     UID...........:  8003
```

Figure 6-23 Installing WebSphere Portal into secondary node - Configuring base portal into secondary node (2 of 3)

Panel 3 of 3, shown in Figure 6-24 on page 192, prompts for all of the groups and user IDs that are used by WebSphere Application Server. These IDs and groups must be defined to RACF and given a USS UID or GID.

In our case, we used the default keyring that gets generated with the WebSphere Application Server installation.



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

 Review WebSphere Application Server customization variables
  Security Domain Configuration (3 of 3)

   Review the following WebSphere Application Server definitions
   and modify where required. Then press ENTER to continue.


 WebSphere Application Server Configuration Group Information
   Group....:  WPCFG          GID..:  8500

 WebSphere Application Server Administrator Information
   User ID..:  WPADMIN        UID..:  8000
   Password.:  WPADMIN

 WebSphere Application Server Servant Group Information
   Group....:  WPSRG          GID..:  8501

 SSL Customization
   Certificate authority keylabel:  WebSphereCA.WP
   Default RACF keyring name.....:  WPKeyring
```

Figure 6-24 Installing WebSphere Portal into secondary node - Configuring base portal into secondary node (3 of 3)

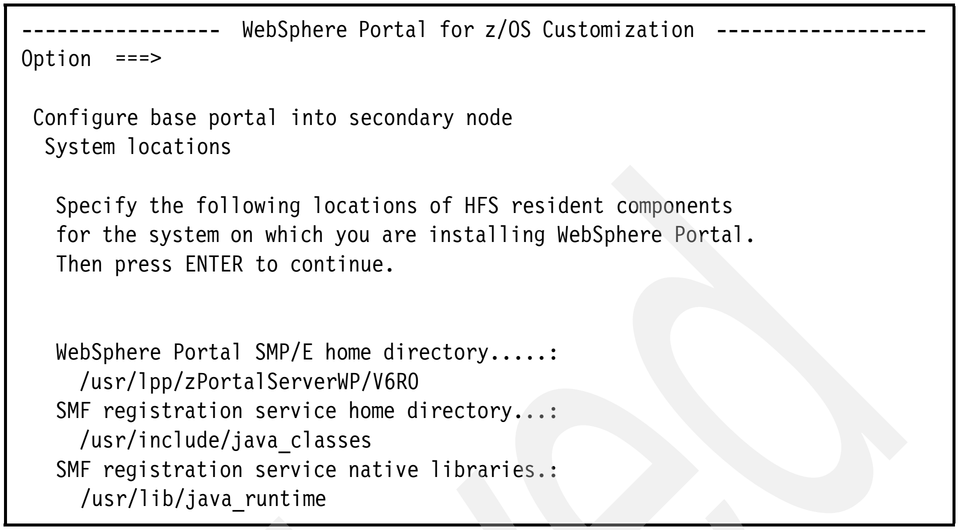
Press F3 to return to the main menu (shown in Figure 2-30 on page 67). From that menu, select option **2 - System locations (directories, HLQs, etc)** as shown:

```
2 - System locations (directories, HLQs, etc)
```

The panel that is displayed, shown in Figure 6-25 on page 193, requests information about the USS directories used on the system. The first value is the USS directory where SMP/E copied the WebSphere Portal code.

`SMF registration service home directory` can be found by searching the HFS for files ifaedjreg.jar and ifaedregDoc.jar.

`SMF registration service native libraries` can be found by searching the HFS for libifaedjreg.so and libifaedjreg64.so.



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

 Configure base portal into secondary node
  System locations

   Specify the following locations of HFS resident components
   for the system on which you are installing WebSphere Portal.
   Then press ENTER to continue.


   WebSphere Portal SMP/E home directory.....:
     /usr/lpp/zPortalServerWP/V6R0
   SMF registration service home directory...:
     /usr/include/java_classes
   SMF registration service native libraries.:
     /usr/lib/java_runtime
```

Figure 6-25 Installing WebSphere Portal into secondary node - Configuring base portal into secondary node (system locations)

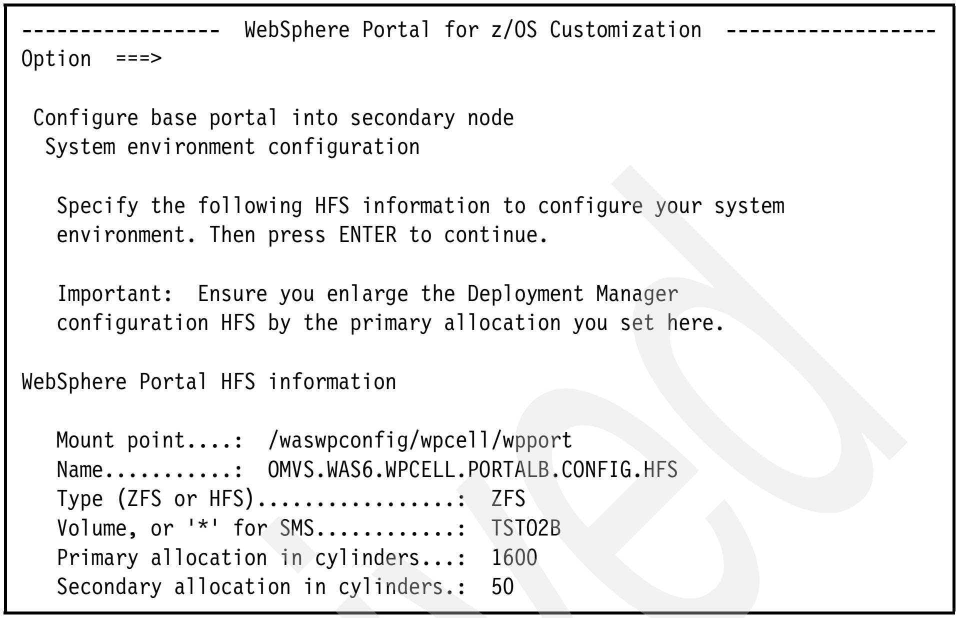
Press F3 once again to return to the main menu (shown in Figure 2-30 on page 67). From that menu, this time select option **3 - System environment configuration** as shown:

```
3 - System environment configuration
```

In the panel that is displayed, shown in Figure 6-26 on page 194), enter the information about the WebSphere Portal HFS that is created by a batch job in a later step. For convenience, the mount point should be under the cell name in the directory structure.

`Name` is the HFS dataset name that is defined for WebSphere Portal. It will be mounted at the mount point variable. We chose to use zFS for the data set type to match what was done during the initial install in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35. The zFS needs to be defined with at least 1600 cylinders, as described in Chapter 6, “Implementing high availability with WebSphere Portal on z/OS” on page 165.

Make sure the volumes that are entered have enough space to accommodate the primary allocation. If you enter multiple volumes, use a comma (,) to separate them. If the volumes are SMS-managed, however, use an asterisk (*) to separate them.



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

 Configure base portal into secondary node
  System environment configuration

   Specify the following HFS information to configure your system
   environment. Then press ENTER to continue.

   Important:  Ensure you enlarge the Deployment Manager
   configuration HFS by the primary allocation you set here.

WebSphere Portal HFS information

   Mount point....:  /waswpconfig/wpcell/wpport
   Name...........:  OMVS.WAS6.WPCELL.PORTALB.CONFIG.HFS
   Type (ZFS or HFS)................:  ZFS
   Volume, or '*' for SMS...........:  TST02B
   Primary allocation in cylinders...:  1600
   Secondary allocation in cylinders.:  50
```

Figure 6-26   System environment configuration

Press F3 again to return to the main menu (shown in Figure 2-30 on page 67). From that menu, this time select option **4 - System configuration** as shown:

```
4 - Server configuration
```

In the next panel, shown in Figure 6-27 on page 195, we chose the default Portal home directory called /Portal, which is appended on to the mount point entered in panel 3 - System environment configuration.

`Node host name` is the host name of the secondary LPAR. Do not use the DVIPA name if using Sysplex Distributor.

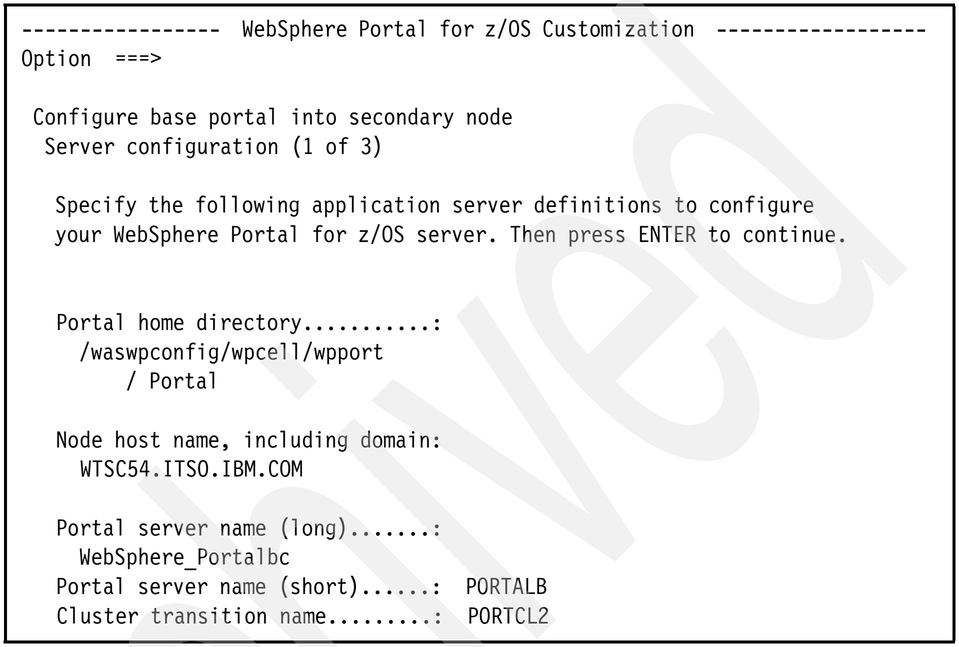
`Portal server name (long)` is the name that is used in the WebSphere Application Server administrative console for the WebSphere Portal application server on the secondary node.

`Portal server name (short)` is the name that is given to the WebSphere Portal started task in SDSF.

`Cluster transition name` is the name used for the process that the application server control and servant region use to communicate.

Do not set the server long name to the name that you ultimately want to use for the Portal in the secondary node. The name you specify here will be used in the

EJPSNDC2 job to create a temporary server that will be deleted during the clustering process later. If you specify the long name that you want to use at this stage, you can not choose that same name for the cluster member in the secondary node. In our case we specified WebSphere_Portalbc here, but we intend to use WebSphere_Portalb as the cluster member name.



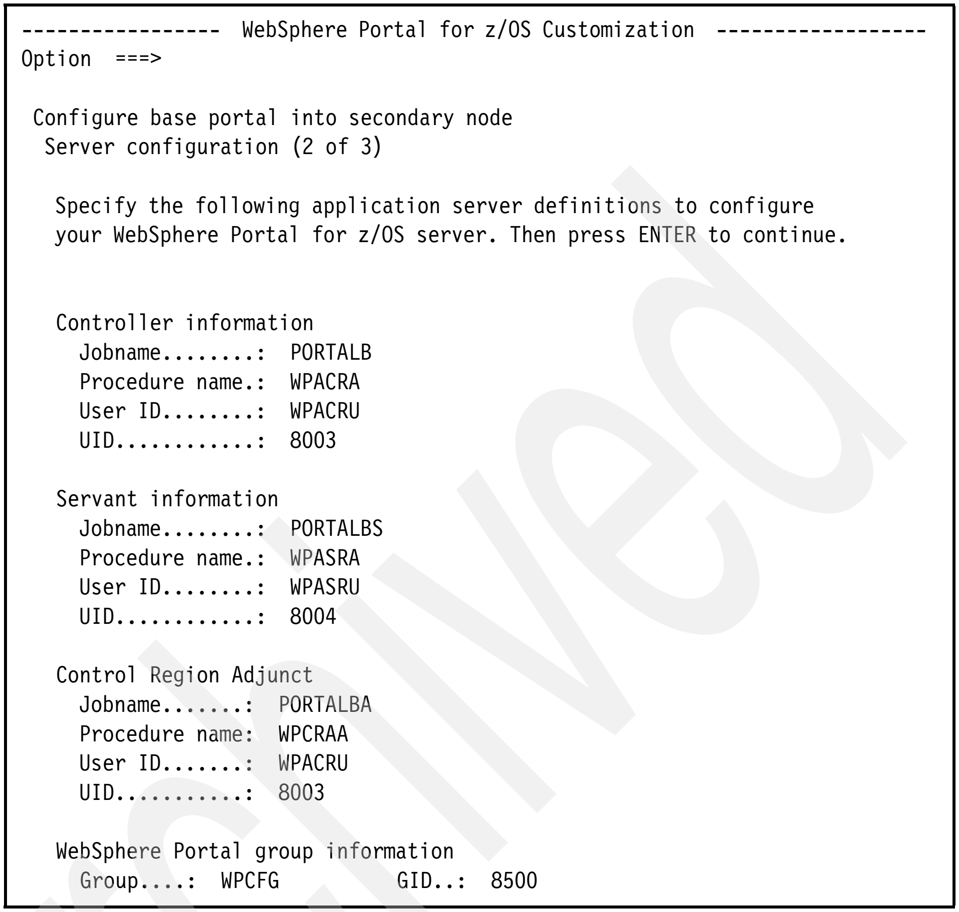
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Configure base portal into secondary node
  Server configuration (1 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   Portal home directory...........:
     /waswpconfig/wpcell/wpport
         / Portal

   Node host name, including domain:
     WTSC54.ITSO.IBM.COM

   Portal server name (long).......:
     WebSphere_Portalbc
   Portal server name (short)......:  PORTALB
   Cluster transition name.........:  PORTCL2
```

Figure 6-27 Installing WebSphere Portal into secondary node - Configure base portal into secondary node (1 of 3)

The jobnames on the next panel (Figure 6-28 on page 196) are generated based on the name given in panel 1 of 3 for `Portal server name (short)`. These jobnames are used for the started task names in SDSF.

`Procedure name` is the name of started task JCL member in SYS1.PROCLIB that is used to start these jobs.

`User ID` is the ID that is used to run the started task. `UID` is the USS UID of the user ID listed. Group is the RACF group that these user IDs are connected to. `GID` is that group’s USS GID.



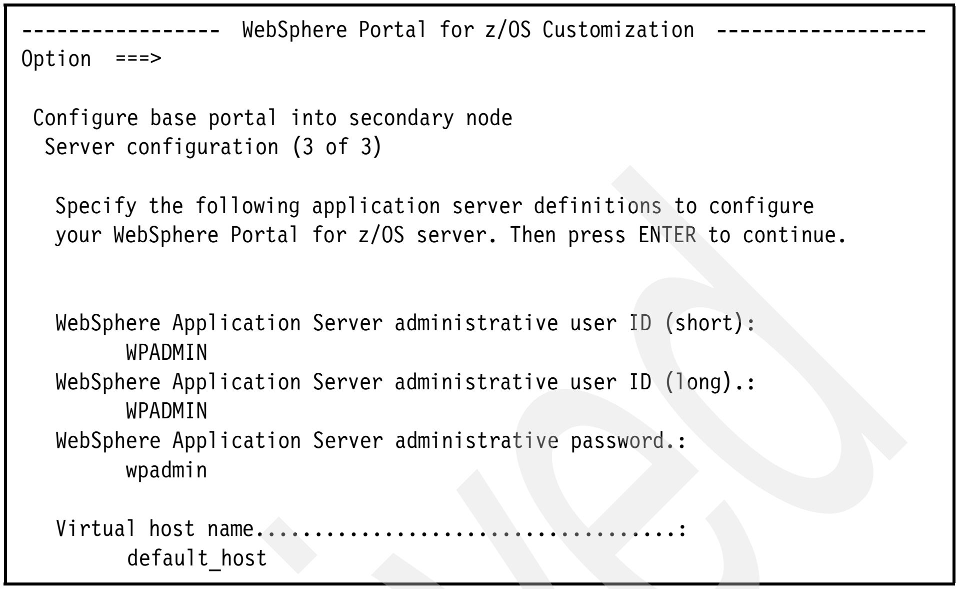
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===>

 Configure base portal into secondary node
  Server configuration (2 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   Controller information
     Jobname........:  PORTALB
     Procedure name.:  WPACRA
     User ID........:  WPACRU
     UID............:  8003

   Servant information
     Jobname........:  PORTALBS
     Procedure name.:  WPASRA
     User ID........:  WPASRU
     UID............:  8004

   Control Region Adjunct
     Jobname.......:  PORTALBA
     Procedure name:  WPCRAA
     User ID.......:  WPACRU
     UID...........:  8003

   WebSphere Portal group information
     Group....:  WPCFG          GID..:  8500
```

Figure 6-28 Installing WebSphere Portal into secondary node - Configure base portal into secondary node (2 of 3)

The panel shown in Figure 6-29 on page 197 requests the user ID that is defined to RACF which is the WebSphere Application Server administrative ID. The password must also be entered here and is case-sensitive.

`Virtual host name` is the name of the WebSphere Application Server virtual host where the WebSphere Portal application ports are defined.

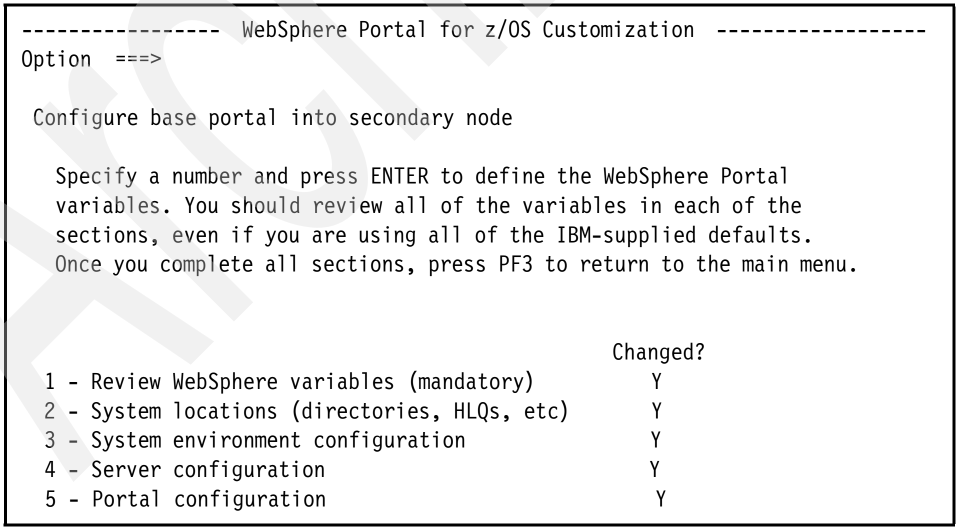
Port `9080` and `9443` are the default virtual host ports for WebSphere Portal, but we changed them to meet our system requirements.



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

 Configure base portal into secondary node
  Server configuration (3 of 3)

   Specify the following application server definitions to configure
   your WebSphere Portal for z/OS server. Then press ENTER to continue.


   WebSphere Application Server administrative user ID (short):
         WPADMIN
   WebSphere Application Server administrative user ID (long).:
         WPADMIN
   WebSphere Application Server administrative password.:
         wpadmin

   Virtual host name...................................:
         default_host
```

Figure 6-29 Installing WebSphere Portal into secondary node - Configure base portal into secondary node (3 of 3)

All of the variables have now been changed, as shown in Figure 6-30. The next step is to generate jobs.



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

 Configure base portal into secondary node

   Specify a number and press ENTER to define the WebSphere Portal
   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 - Review WebSphere variables (mandatory)           Y
  2 - System locations (directories, HLQs, etc)        Y
  3 - System environment configuration                 Y
  4 - Server configuration                             Y
  5 - Portal configuration                             Y
```

Figure 6-30 Base portal variables completed

Press F3 to return to the main menu once again (see Figure 2-31 on page 68). From there, select option **3 - Generate customization jobs. Validate your customization variables and generate jobs and instructions** to generate the batch jobs.

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

After the jobs are generated, install instructions can be viewed by selecting option **4**, or by going to CNTL member EJPINDS.

```
4  View instructions. View the generated customization instructions.
```

Running the WebSphere Portal batch jobs

Installing the WebSphere Portal base code into the secondary node generates many of the same jobs that the primary node install created in Chapter 6, “Implementing high availability with WebSphere Portal on z/OS” on page 165.

EJPSRACF

This job gives out authority to RACF resources to the IDs that were entered into the ISPF dialog. We do not run this job during our secondary node install because we used the same IDs and many of the same started task names.

We chose to manually run three of the RACF statements that this job runs. The three statements were isolated, because we used a new application server name for WebSphere Portal on the secondary node. All of the other RACF statements were previously run with the primary node WebSphere Portal install and we did not want to run them again.

```
RDEFINE STARTED WPACRA.PORTALB  STDATA(USER(WPACRU) +
 GROUP(WPCFG) TRACE(YES))
RDEFINE STARTED PORTALBS.* STDATA(USER(WPASRU) +
 GROUP(WPCFG) TRACE(YES))
RDEFINE STARTED PORTALBA.* STDATA(USER(WPACRU) +
 GROUP(WPCFG) TRACE(YES))
```

EJPSCHFS

This job creates the mount point directory for the portal’s configuration HFS and mounts the dataset to the mount point. Make sure this job is run with an ID that has a UID of 0, so that it has sufficient authority to create files and directories. This ID must also have an OMVS segment so that it can write the installation log to this directory structure. Set the permissions of the portal mount point to 775 so that the WebSphere Application Server administrative user and group can write to this directory structure.

EJPSCFG1
Before running this job, start the Deployment Manager and Node Agent. This job performs an archive install of the WebSphere Portal Server configuration files.

In order to complete this, the Portal install program needs access to a zip and unzip utility. Free versions of the zip and unzip utilities can be found at the following site:

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty1.html

After the utilities are downloaded, the .profile of the user running the job must point to the location of the utilities so that the job knows where to find them. The line added to .profile should look like this:

```
export PATH=$PATH:/ziptool/utility/path/
```

EJPSNDC2
This job configures the base WebSphere Portal into the secondary node of the WebSphere Deployment Manager cell. Make sure all the directories and subdirectories are owned by the WebSphere administrator ID and group (WPADMIN:WPCFG) and that the permissions are set to 775. If the ownership or permissions are wrong, the job will fail since it will not be able to create any HFS objects such as files or directories..

6.2.5 Cluster Portal on the primary node

In our scenario, we have two WebSphere Application Server nodes defined currently which are being managed by one Deployment Manager. Each node has one application server defined to it that is being used for WebSphere Portal.

The next step in the construction of a high availability environment is to create a cluster which will contain the two application servers. The first task in creating a cluster is to set the first application server to be a member of the cluster. Later, this member will be cloned on the secondary node and added to the cluster.

Customizing the ISPF dialog

Like many of the previous tasks, a cluster is prepared via the ISPF dialog. Launch the ISPF dialog on the primary WebSphere Application Server LPAR, as follows:

```
exec 'BBWP6049.SBBOCLIB(BBOWSTRT)' 'APPL(PS1) PROD(EJP)
PRODHLQ(BBWP6049)'
```

> **Note:** Because we returned to configuring the primary node, the APPL parameter was set to APPL(PS1).

After the dialog starts, select option **5**:

```
5  WebSphere Application Server-based add-on products. Configure
   other products that are built on WebSphere Application Server.
```

Next choose option 1 for WebSphere Portal:

```
1   WebSphere Portal for z/OS
    Configure WebSphere Portal
```

Choose **1 - Basic configuration tasks**:

```
1  Basic configuration tasks. If you want to configure
   a base portal, use this option
```

Select option **2**:

```
2  Configure base portal into a WebSphere Network Deployment cell.
   If you want to configure a base portal into a WebSphere network
   deployment cell, use this option.  This selection includes
   options for configuring a portal with a Cloudscape database or a
   DB2 for z/OS database.
```

To set up the portal cluster, select option **4**:

```
4  Configure portal cluster.
   Select this option to create and configure a portal cluster in
   your WebSphere Network Deployment cell.
```

Load the customization variables used in last task that you performed against the primary node by choosing option **L**. The variables to load should be those that you saved (by using SAVED) when you performed the database transfer.

```
L  Load customization variables. Load your customization variables
from a data set.
```

Define the data sets where the customization jobs are stored by choosing option **1**:

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

Set all the variables for the WebSphere Portal cluster with option **2**:

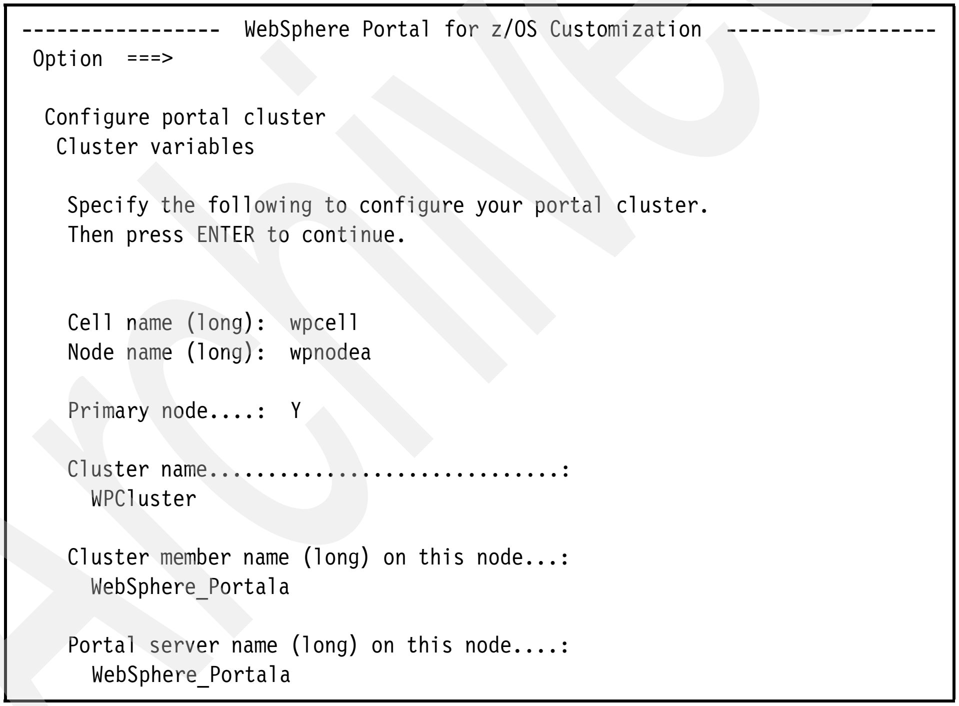
```
2  Define variables. Define your installation-specific information
   for your WebSphere Portal cluster.
```

There is only one panel with variables for the cluster creation task, as shown in Figure 6-31 on page 201. The Deployment Manager cell name and node name are already filled in based on previous configuration, and cannot be changed here. Indicate that this is the primary node by setting `Primary node` to `Y`.

`Cluster name` is the name that the cluster will be given within WebSphere Application Server, and it will be displayed in the administrative console under **Servers** -> **Clusters**.

`Cluster member name` is the name of the application server on the primary node that is the first member of the cluster. In our case, we gave the Cluster member name the same name as the Portal server name.

On the primary node, the cluster member name must be the same as the portal server name on this node. If you specify a different name, the dialog issues a message `Invalid cl.member name`. Note that Portal server name cannot be changed on this panel.



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

  Configure portal cluster
   Cluster variables

    Specify the following to configure your portal cluster.
    Then press ENTER to continue.


    Cell name (long):  wpcell
    Node name (long):  wpnodea

    Primary node....:  Y

    Cluster name.............................:
      WPCluster

    Cluster member name (long) on this node...:
      WebSphere_Portala

    Portal server name (long) on this node....:
      WebSphere_Portala
```

Figure 6-31 Configure Portal cluster - Cluster variables

After all the variables are entered, generate the customization job by selecting option **3**:

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

Only generate the jobs for this task. Select option **1**:

```
1  Generate customization jobs for this task. If you want to
generate jobs for this customization task only, use this option.
```

Select option **4** to view the customization instructions or look at CNTL member EJPINDS.

```
4  View instructions. View the generated customization instructions.
```

Running the batch job to create the cluster and verify creation

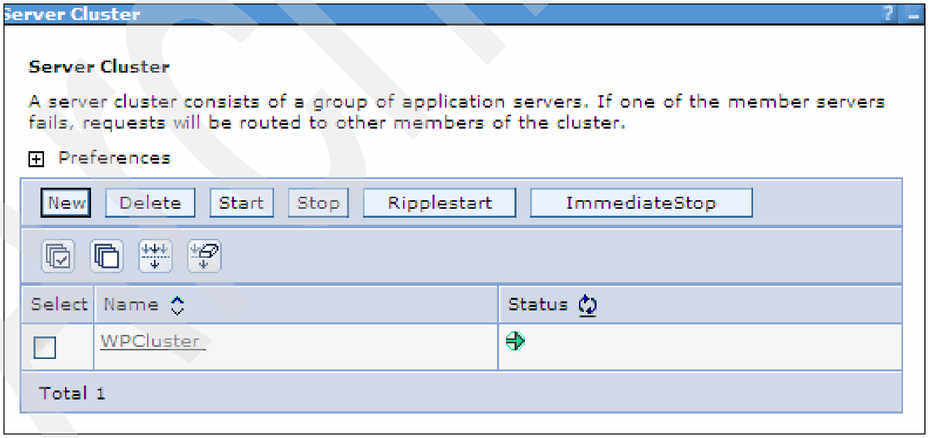
There is only one batch job required to create the cluster on the primary WebSphere Application Server node.

EJPSNDS

Stop the WebSphere Portal application server before running this job. EJPSNDS creates the portal cluster on the primary node with the one application server as a member.

After EJPSNDS completes with a RC=0, use the WebSphere Application Server administrative console to verify the existence of the cluster. Navigate to **Servers -> Clusters** (see Figure 6-32).

There is one cluster defined and it is in stopped status. Start the cluster by selecting the check box and clicking **Start**. Starting the cluster also starts all of the application servers connected to the cluster.



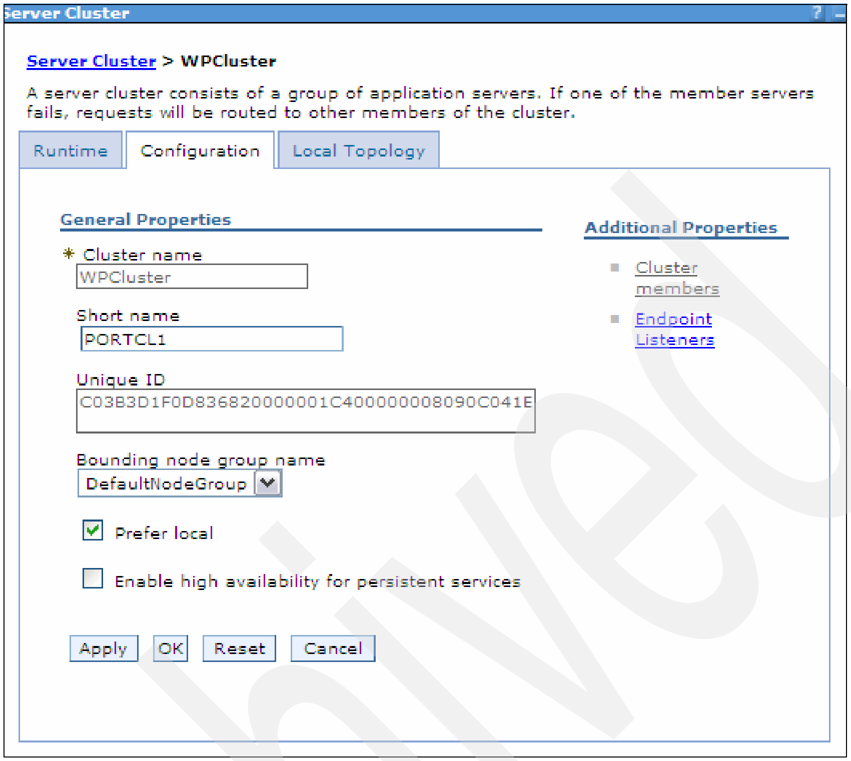
Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Select	Name	Status
□	WPCluster	
Total 1		

Figure 6-32 Started cluster

Click the cluster name to display more of the cluster properties (see Figure 6-33 on page 203).



Server Cluster

Server Cluster > WPCluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

General Properties

* Cluster name
WPCluster

Short name
PORTCL1

Unique ID
C03B3D1F0D836820000001C400000008090C041E

Bounding node group name
DefaultNodeGroup

☑ Prefer local

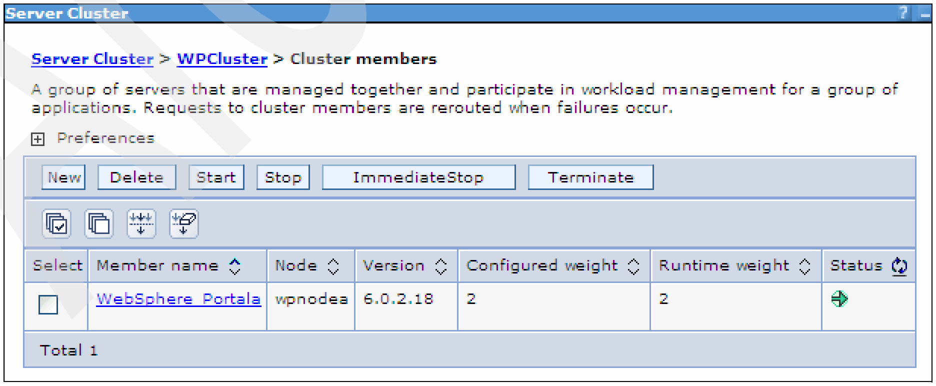
□ Enable high availability for persistent services

Additional Properties

- Cluster members
- Endpoint Listeners

Figure 6-33 The cluster, with three presentation options

From this screen, select Cluster members to view all of the application servers that are connected to the cluster (see Figure 6-34). Notice that only the primary node application server is connected to the cluster at this time.



Server Cluster

Server Cluster > WPCluster > Cluster members

A group of servers that are managed together and participate in workload management for a group of applications. Requests to cluster members are rerouted when failures occur.

⊞ Preferences

Select	Member name	Node	Version	Configured weight	Runtime weight	Status
□	WebSphere_Portala	wpnodea	6.0.2.18	2	2	➔
Total 1						

Figure 6-34 Cluster members

6.2.6 Configuring the Portal cluster in the secondary node

The next step in the process of building the cluster is to add a new cluster member on the secondary node to the cluster.

For more information about this step, refer to the following sections in the WebSphere Portal for z/OS InfoCenter:

- “Installing secondary nodes” at:
  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/tins_zosndsec_zos.html`
- “Configure base Portal in secondary node” at:
  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/cu_cdndsecnode_zos.html`

Running the ISPF dialog

Start the WebSphere Application Server ISPF dialog on the secondary LPAR, as follows:

```
exec 'BBWP6054.SBBOCLIB(BBOWSTRT)' 'APPL(PS2) PROD(EJP)
PRODHLQ(BBWP6054)'
```

Note: Because we were now working on the secondary node, the APPL parameter was set to APPL(PS2).

After the dialog starts, select option **5**:

```
5  WebSphere Application Server-based add-on products. Configure
   other products that are built on WebSphere Application Server.
```

Choose option 1 for WebSphere Portal:

```
1   WebSphere Portal for z/OS
    Configure WebSphere Portal
```

Choose **1 - Basic configuration tasks**:

```
1  Basic configuration tasks. If you want to configure
   a base portal, use this option
```

Select option **2**:

```
2  Configure base portal into a WebSphere Network Deployment cell.
   If you want to configure a base portal into a WebSphere network
   deployment cell, use this option.  This selection includes
   options for configuring a portal with a Cloudscape database or a
   DB2 for z/OS database.
```

Enter option **4** to configure the portal cluster.

```
4  Configure portal cluster.
    Select this option to create and configure a portal cluster in
    your WebSphere Network Deployment cell.
```

Select **L** to load the custom variables used in previous tasks. The variables you should load are those that you saved when you configured the base Portal into the secondary node.

```
L  Load customization variables. Load your WebSphere Portal
   customization variables from a data set.
```

Select option **1** to verify the dataset where the install jobs will get generated. If no changes are made, the jobs generated into the same dataset that was used for all the other installation tasks.

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

Choose **2** to define the variables for the configure portal cluster task.

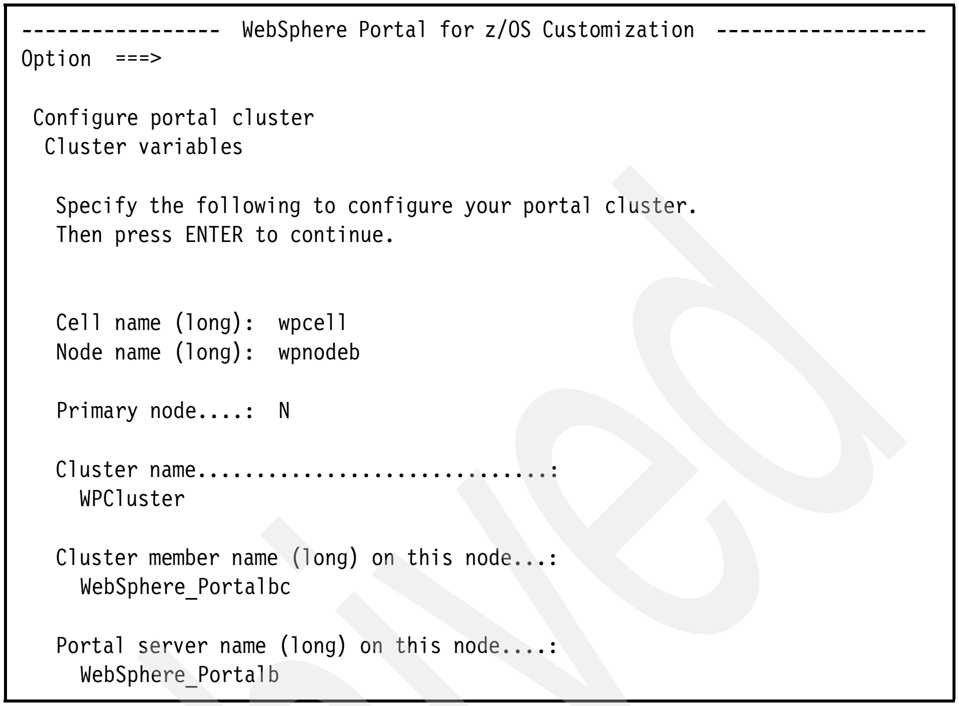
```
2 Define variables. Define your installation-specific information
      for your WebSphere Portal cluster.
```

In the panel shown in Figure 6-35 on page 206, `Cell name`, `Node name` and `Portal server name` are set values that cannot be changed on this screen. 6.2.5, “Cluster Portal on the primary node” on page 199 describes how to create the cluster in the primary node. Because this is the secondary node, set `Primary node` to `N`.

Use the same cluster name that was defined in 6.2.5, “Cluster Portal on the primary node” on page 199. Set `Cluster member name (long)` to the final name you want the Portal server to have in the secondary node.

`Portal server name (long) on this node` should already be populated with the “temporary” server name that was specified when the base Portal was configured in the secondary node.

Note that, on secondary nodes, the cluster member name must be different to the portal server name on the node. If you choose the same name, the dialog issues the message `Invalid cl.member name`.



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

 Configure portal cluster
  Cluster variables

   Specify the following to configure your portal cluster.
   Then press ENTER to continue.


   Cell name (long):  wpcell
   Node name (long):  wpnodeb

   Primary node....:  N

   Cluster name.............................:
     WPCluster

   Cluster member name (long) on this node...:
     WebSphere_Portalbc

   Portal server name (long) on this node....:
     WebSphere_Portalb
```

Figure 6-35 Configure Portal cluster - Cluster variables

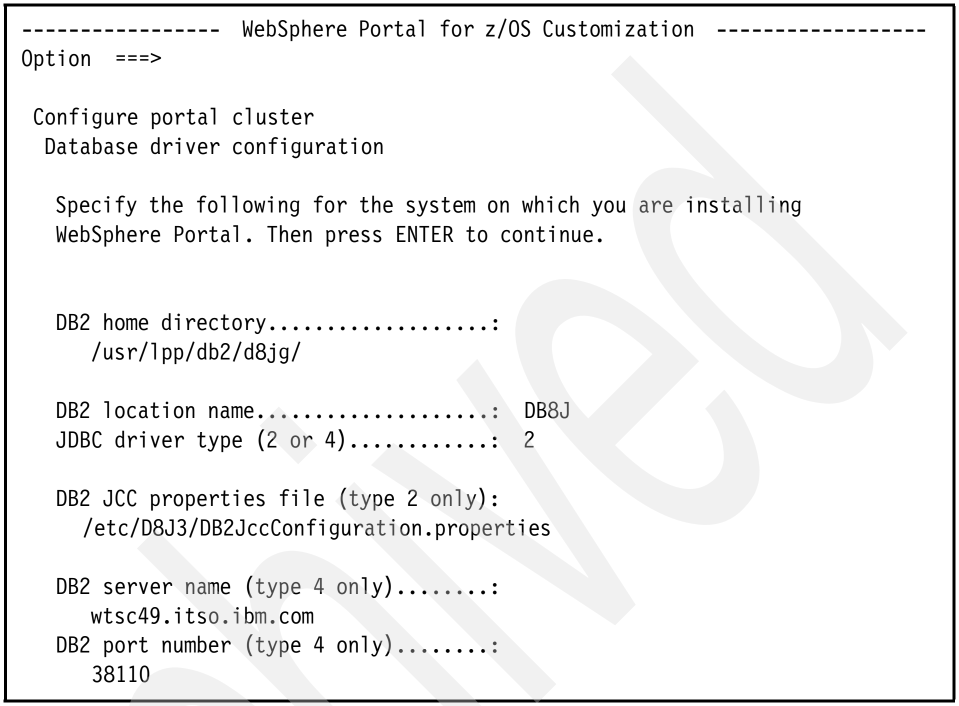
The next panel, as shown in Figure 6-36 on page 207, describes the DB2 information used for the WebSphere Portal database. We used a DB2 datasharing group for our DB2s.

`DB2 home directory` is the USS home directory for the DB2 on the secondary LPAR. `DB2 location name` is the DDF location name of the DB2 on the secondary LPAR.

A JDBC Type 2 driver is being used. When using a JDBC Type 2 driver, a DB2JccConfiguration.properties file has to be manually created to tell the driver the subsystem ID of the DB2 to be used. Our DB2JccConfiguration.properties file was stored in /etc/D8J3/. There was only one line in our file:

```
db2.jcc.ssid=D8J3
```

In our case, we filled in the fields DB2 server name and DB2 port number because they are used by one of the batch jobs, but not used by the WebSphere Portal product.



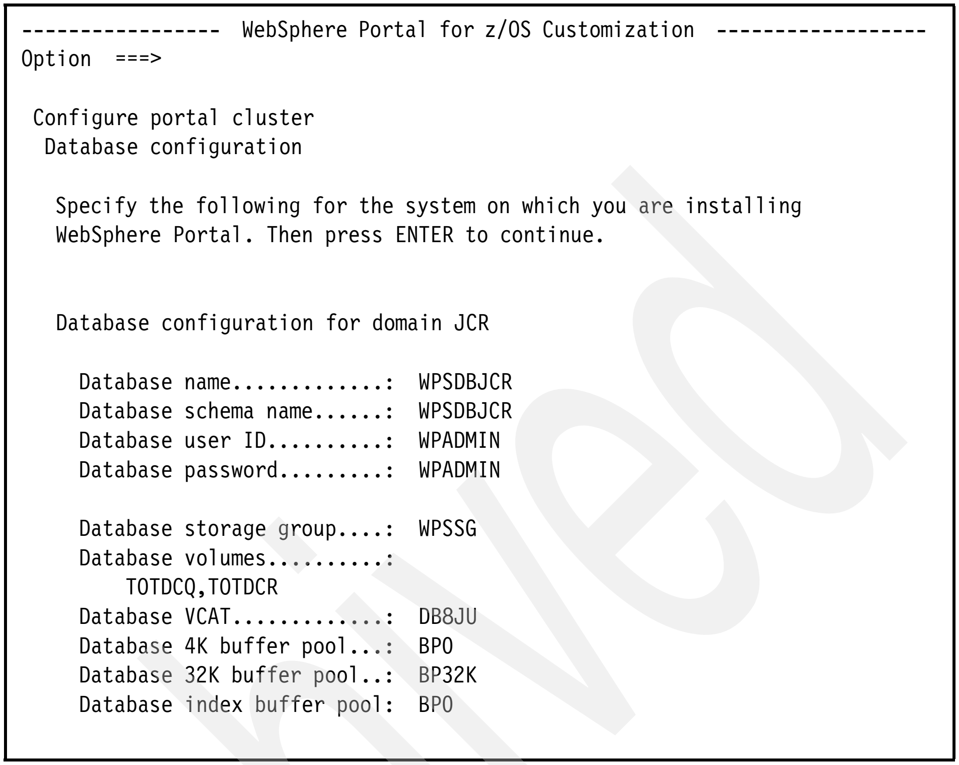
```
----------------  WebSphere Portal for z/OS Customization  -----------------
Option  ===>

 Configure portal cluster
  Database driver configuration

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   DB2 home directory..................:
      /usr/lpp/db2/d8jg/

   DB2 location name...................:  DB8J
   JDBC driver type (2 or 4)...........:  2

   DB2 JCC properties file (type 2 only):
     /etc/D8J3/DB2JccConfiguration.properties

   DB2 server name (type 4 only)........:
      wtsc49.itso.ibm.com
   DB2 port number (type 4 only)........:
      38110
```

Figure 6-36 Configure Portal cluster - Database driver configuration

The following panel, shown in Figure 6-37 on page 208, asks for the information about the WebSphere Portal database. The database name and schema name values are set to the database name used to store WebSphere Portal data.

The database user ID and database password variables are set to the ID that created the WebSphere Portal database in DB2, or any other ID that has database administrator access over this database.

The database storage group and database volumes information can be obtained from the storage group definition in DB2. Database VCAT is the high level qualifier of the data sets used for the DB2 databases. This panel also requests the buffer pools used in DB2 for the WebSphere Portal objects.



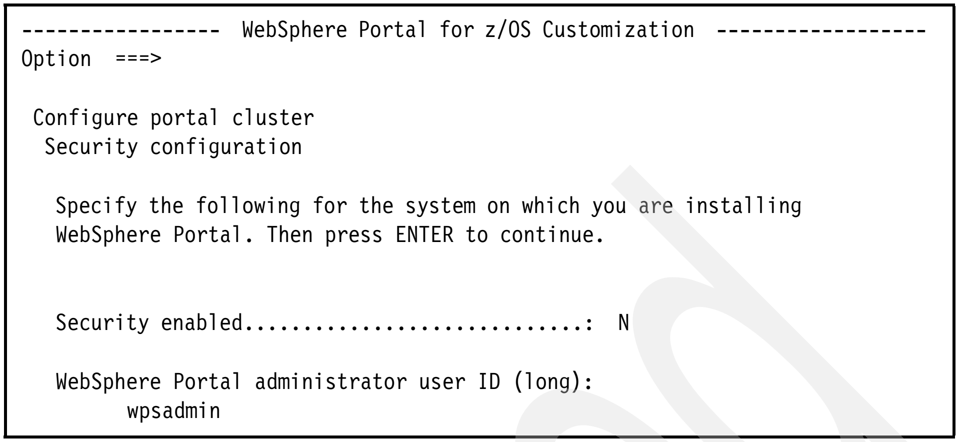
```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

 Configure portal cluster
  Database configuration

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Database configuration for domain JCR

     Database name.............:  WPSDBJCR
     Database schema name......:  WPSDBJCR
     Database user ID..........:  WPADMIN
     Database password.........:  WPADMIN

     Database storage group....:  WPSSG
     Database volumes..........:
         TOTDCQ,TOTDCR
     Database VCAT.............:  DB8JU
     Database 4K buffer pool...:  BP0
     Database 32K buffer pool..:  BP32K
     Database index buffer pool:  BP0
```

Figure 6-37 Database configuration

Important: The values entered for the database configuration must be the same as the values for the JCR domain on the primary node.

As you may remember, we disabled security in 6.2.2, “Disabling global security” on page 168, so here, set `Security enabled` to `N` in the panel shown in Figure 6-38 on page 209.

Note that `wpsadmin` is the WebSphere Portal administrator that we entered when the original portal was built.



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

 Configure portal cluster
  Security configuration

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   Security enabled.............................:  N

   WebSphere Portal administrator user ID (long):
         wpsadmin
```

Figure 6-38 Configure Portal cluster - Security configuration

All the variables have now been entered. Select option **3** to generate the batch job for this task.

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

Because you only want to generate the job for the portal cluster on the secondary node, select option **1**.

```
1  Generate customization jobs for this task. If you want to
   generate jobs for this customization task only, use this option.
```

Running the batch customization job

Only one job gets generated for adding the secondary application server to the WebSphere Portal cluster.

EJPSNDS

This job adds a new cluster member on the secondary node, and deletes the portal server on the secondary node. Make sure EJPSNDS runs with the WebSphere Application Server administrator ID in the jobcard.

After EJPSNDS has completed with a RC=0, you need to correct the server short name in the secondary node's cluster member because it will have been given the default short name of BBOS001. Follow these steps:

1. Navigate to **Servers -> Application Servers** and then click the cluster member in the secondary node, WebSphere_Portalb. Change the server short name from BBOS001 to PORTALB, then click **OK**.

2. Verify that the cluster now includes a cluster member on the secondary node. Go to the WebSphere Administrative Console and select **Servers** -> **Clusters**. Select the check box next to the cluster name and click **start**.

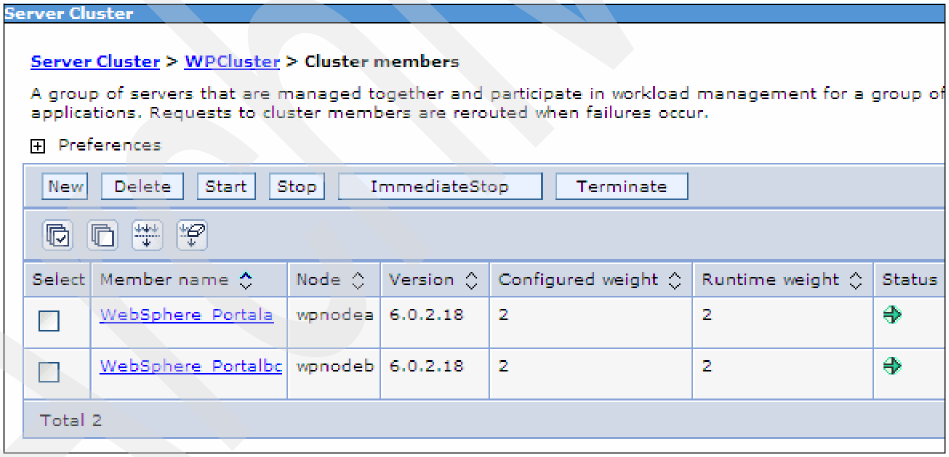
   Make sure that both application servers get started by starting the cluster. Verify that the WebSphere Portal address space gets started successfully on both LPARs.

   The error shown in Example 6-1 may occur if a Sysplex Distributor is being used and the sysplex name was defined to one of the application servers instead of the LPAR name.

Example 6-1 Possible error with Sysplex Distributor

```
./bbooreq.cpp+554 ... BBOO010W The function
CORBA::Request::Request(CORBA::Object_ORBProxy_ptr, char *,
CORBA::Flags)+554 raised CORBA system exception CORBA::COMM_FAILURE.
Error code is C9C21149.
```

Verify that the cluster has both members defined to it by going to **Servers -> Clusters ->** ***Cluster_name*** **-> Cluster members** in the WebSphere administrative console, as shown in Figure 6-39.



Server Cluster

Server Cluster > WPCluster > Cluster members

A group of servers that are managed together and participate in workload management for a group of applications. Requests to cluster members are rerouted when failures occur.

Select	Member name ⇕	Node ⇕	Version ⇕	Configured weight ⇕	Runtime weight ⇕	Status
□	WebSphere_Portala	wpnodea	6.0.2.18	2	2	➔
□	WebSphere_Portalbc	wpnodeb	6.0.2.18	2	2	➔
Total 2						

Figure 6-39 WPCluster members

The diagram shown in Figure 6-40 on page 211 shows the environment that is now set up. There are two nodes federated into the Deployment Manager. Each node has its own application server for WebSphere Portal.

PORTALA is the application server on the primary node. PORTALB is the application server on the secondary node.

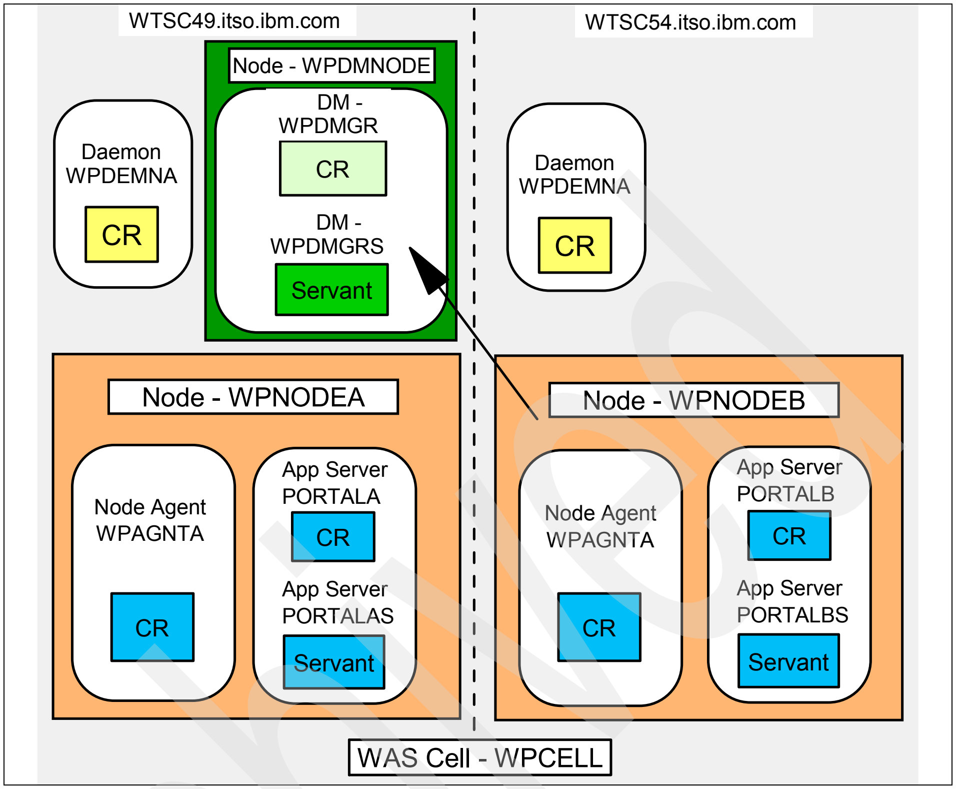


Figure 6-40 Network Deployment diagram with two nodes and two Portal application servers

6.2.7 Reenabling security on primary node and secondary node

This first step completed in this chapter was to turn off global security in WebSphere Application Server. We turned off security to make completing the other steps in creating the cluster easier. Global security often adds complexity to the clustering jobs which can be avoided by simply temporarily turning off security. Now that the cluster is built, however, we can turn security back on.

Remember that we had enabled security in the Portal server by running job EJPSSER. That job enabled Portal security using an LDAP server and with support for realms.

Although the Portal cluster member in the secondary node was cloned from the Portal in the primary node, not all of the Portal configuration is held within WebSphere Application Server xml files. Therefore, the process of cloning the Portal server has not completely enabled Portal security in the secondary node.

For this reason, you need to run the Portal ISPF dialogs again to completely enable Portal security in the secondary node.

Enabling security in the Deployment Manager

Start by logging on to the WebSphere administrative console. Set global security on in the Deployment Manager by selecting **Security -> Global security** and checking the Enable global security' check box.

Click **Apply** and save the changes. Recycle the Deployment Manager and Node Agent on both LPARs to pick up the changes. Leave the Portal application servers down while completing the next step.

Enabling Portal security in the secondary node

Start the WebSphere Application Server ISPF dialog on the secondary LPAR, as follows:

```
exec 'BBWP6054.SBBOCLIB(BBOWSTRT)' 'APPL(PS2) PROD(EJP)
PRODHLQ(BBWP6054)'
```

After the dialog starts, select option **5**:

```
5  WebSphere Application Server-based add-on products. Configure
   other products that are built on WebSphere Application Server.
```

Choose option **1** for WebSphere Portal:

```
1   WebSphere Portal for z/OS
    Configure WebSphere Portal
```

Select option **3 - Security configuration tasks**:

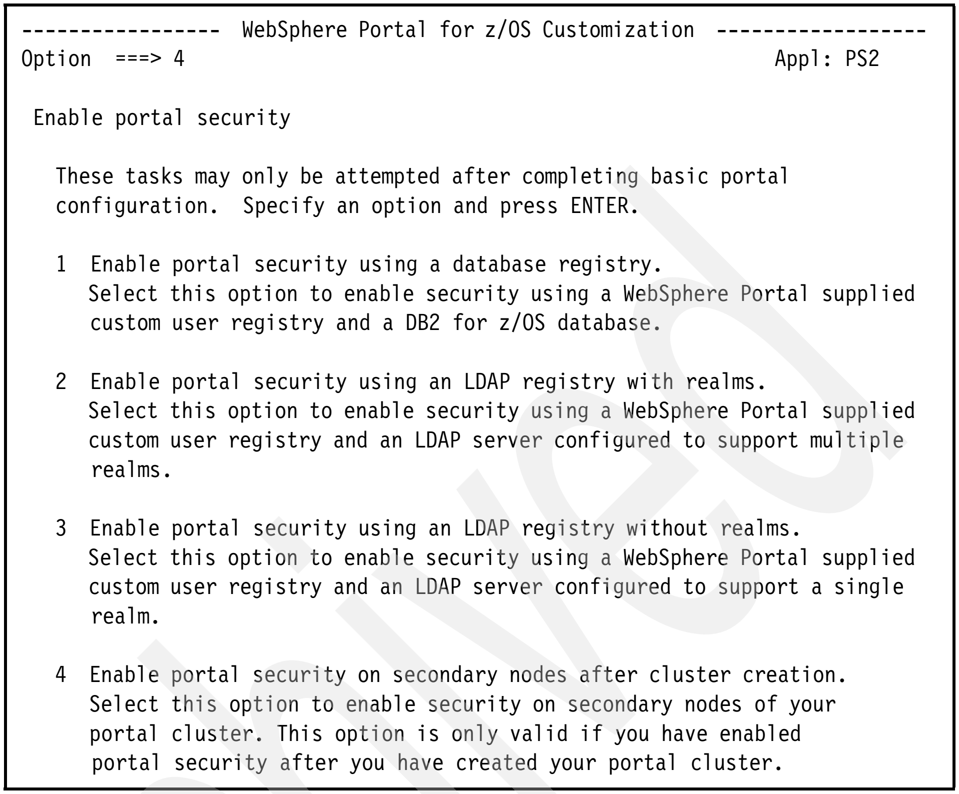
```
3  Security configuration tasks. If you want to configure security
   for your portal, use this option.
   You must complete option 1 before starting this option.
```

Select option **1** to enable portal security:

```
1  Enable portal security. Select this option to enable security for
   your portal.  This option is only valid if security is currently
   disabled for your portal.
```

We already configured security for WebSphere Portal on our primary node in Chapter 4, "Securing WebSphere Portal with LDAP" on page 109 using option 2.

Because we already have one Portal application server with security enabled, select option **4** to secure the Portal on the clustered secondary node, as shown in Figure 6-41 on page 213.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 4                                                   Appl: PS2

 Enable portal security

   These tasks may only be attempted after completing basic portal
   configuration.  Specify an option and press ENTER.

   1  Enable portal security using a database registry.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and a DB2 for z/OS database.

   2  Enable portal security using an LDAP registry with realms.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and an LDAP server configured to support multiple
      realms.

   3  Enable portal security using an LDAP registry without realms.
      Select this option to enable security using a WebSphere Portal supplied
      custom user registry and an LDAP server configured to support a single
      realm.

   4  Enable portal security on secondary nodes after cluster creation.
      Select this option to enable security on secondary nodes of your
      portal cluster. This option is only valid if you have enabled
      portal security after you have created your portal cluster.
```

Figure 6-41 Enable Portal security

Load the customization variables using option **L**. Load the latest set of saved variables you saved when configuring the base Portal in the secondary node.

```
L  Load customization variables. Load your WebSphere Portal
customization variables from a data set.
```

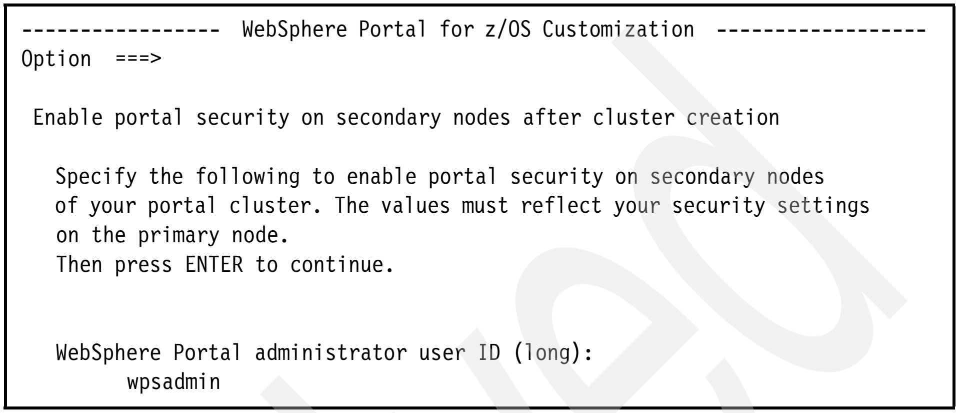
Select option **1** to verify the dataset where the install jobs will get generated. If no changes are made, the jobs are generated into the same dataset that was used for all the other installation tasks.

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

Choose **2** to define the variables for enabling security on the secondary portal application server.

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

The only variable that the ISPF dialog requires is the WebSphere Portal administrator ID that is defined in LDAP. This is the same ID that was used on the primary node in Chapter 4, “Securing WebSphere Portal with LDAP” on page 109.



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

 Enable portal security on secondary nodes after cluster creation

   Specify the following to enable portal security on secondary nodes
   of your portal cluster. The values must reflect your security settings
   on the primary node.
   Then press ENTER to continue.


   WebSphere Portal administrator user ID (long):
         wpsadmin
```

Figure 6-42 Enable Portal security on the secondary node

Select option **3** to generate the job for enabling security on the secondary node:

```
3  Generate customization jobs. Validate your choices
   and generate jobs and instructions.
```

Choose option **1** to only generate the job for the enabling security task:

```
1  Generate customization jobs for this task. If you want to
   generate jobs for this customization task only, use this option.
```

Running the batch job

There is only one batch job that gets generated to enable security on the secondary node. View the instructions for this job by going to option **4**, or look in C NTL member EJPISEC.

```
4  View instructions. View the generated customization instructions.
```

EJPSSEC

This job enables security on the secondary WebSphere Portal application server. EJPSSEC must be run with the WebSphere Application Server administrator ID in the jobcard.

After this job completes with a return code of zero, start both application servers by starting the cluster. Try logging on to each WebSphere Portal page with the WebSphere Portal administrator LDAP ID to verify that security is working properly.

Using search in a cluster

You will probably want to allow users to search content. It is beyond the scope of this document to describe in detail how to configure the search engine, but be aware that when WebSphere Portal is configured in a cluster, the search engine must be deployed in a separate WebSphere Application Server. That server could be running on z/OS, but it cannot be clustered.

Refer to the WebSphere Portal Infocenter article “Configuring search in a clustered environment” for details. The article is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wps/srtcfgsrchclstrenv.html

6.2.8 HTTP plug-in changes for high availability

In Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35, we explain the steps required to generate an HTTP plug-in file that included all the WebSphere Portal applications. However, some changes are required in the HTTP plug-in when a cluster is built.

First, generate the plug-in again by going to **Servers -> Web servers**. Select the Web server and click **Generate Plug-in**. Regenerating the plug-in adds a section to the plugin-cfg.xml for the additional member of the cluster.

In Example 6-2, the SC49 server was created after generating the plug-in in Chapter 2, “WebSphere Portal for z/OS primary node configuration” on page 35. Regenerating the plug-in after building the cluster added the section for SC54.

Example 6-2 The plugin-cfg.xml after clustering

```
- <Server CloneID="PortalaSC49" ConnectTimeout="0" ExtendedHandshake="false"
LoadBalanceWeight="2" MaxConnections="-1" Name="wpnodea_WebSphere_Portala"
ServerIOTimeout="0" WaitForContinue="false">
<Transport Hostname="WTSC49A.ITSO.IBM.COM" Port="29818" Protocol="http" />
- <Transport Hostname="WTSC49A.ITSO.IBM.COM" Port="29819" Protocol="https">
<Property Name="keyring" Value="/opt/IBM/WebSphere/Plugins/etc/plugin-key.kdb" />
<Property Name="stashfile" Value="/opt/IBM/WebSphere/Plugins/etc/plugin-key.sth" />
</Transport>
</Server>
- <Server CloneID="PortalbSC54" ConnectTimeout="0" ExtendedHandshake="false"
LoadBalanceWeight="2" MaxConnections="-1" Name="wpnodeb_WebSphere_Portalbc"
ServerIOTimeout="0" WaitForContinue="false">
<Transport Hostname="WTSC54A.ITSO.IBM.COM" Port="29818" Protocol="http" />
- <Transport Hostname="WTSC54A.ITSO.IBM.COM" Port="29819" Protocol="https">
<Property Name="keyring" Value="/opt/IBM/WebSphere/Plugins/etc/plugin-key.kdb" />
<Property Name="stashfile" Value="/opt/IBM/WebSphere/Plugins/etc/plugin-key.sth" />
</Transport>
```

In most high availability scenarios, multiple HTTP servers are used. When the plug-in is regenerated, the new plugin-cfg.xml must be copied to both HTTP servers to reflect the plug-in changes.

Plug-in configuration with a load balancer

You must make one additional change to the plugin-cfg.xml file if a system is using a load balancer. The cluster address of the load balancer has to be added to the plugin-cfg.xml file.

A *cluster address* is like a server element in that the same attributes and parameters can be specified as a server element. The difference is that only one of them can be defined within a server cluster. Use a cluster address when it is undesirable to have the plug-in perform any type of load balancing. This is the case when there is already some type of load balancer in between the plug-in and the application server. In our case we had a Sysplex Distributor.

If a request comes in that does not have affinity established, the plug-in routes it to the cluster address, if defined. If affinity has been established, then the plug-in routes the request directly to the clone, bypassing the cluster address entirely.

Example 6-3 shows the ClusterAddress section of the plugin-cfg.xml file. The same ports are defined for the ClusterAddress and the application server transports. The only difference is the transport hostname. In our scenario, the hostname of the Sysplex Distributor was haplex1.itso.ibm.com.

Example 6-3 The plugin-cfg.xml with a Sysplex Distributor

```
- <ClusterAddress name="haplex1">
<Transport hostname="HAPLEX1.ITSO.IBM.COM" port="29818" protocol="http" />
<Transport hostname="HAPLEX1.ITSO.IBM.COM" port="29819" protocol="https" />
</ClusterAddress>
```

After the Sysplex Distributor is introduced into the environment, the WebSphere Portal ports have to be defined to the Sysplex Distributor. This is done by editing HLQ.TCPPARMS(TCPPROF).

Example 6-4 shows the section of this file where the WebSphere Portal ports were added. Ports 29818 and 29819 were the ports used by WebSphere Portal on our systems. This file must be updated for the primary Sysplex Distributor and the backup.

Example 6-4 WebSphere Portal TCP/IP ports in Sysplex Distributor

```
VIPADISTRIBUTE DEFINE DISTMETHOD BASEWLM 10.1.6.150
```

```
PORT 9500 9501 9510 9512 9514 9518 9528 9522 9523
     9558 9559 9563 9565 23 9993 9080  80
     7888  39902  39903
     39911 39912 39938 39939 38110
     39910 39913 39915 39918 39919 40008 40009 40108 40109
     40118 40218 40638
     29810 29811 29812 29818 29819 29902 29903
```

6.3 LDAP high availability

WebSphere Portal uses several resources that need to be considered when discussing a high availability environment. Earlier in this chapter we discuss setting up a cluster to make sure WebSphere Application Server is highly available. The DB2 database is required for WebSphere Portal to function. DB2 high availability is done through DB2 datasharing, which is not covered in this book. Finally, WebSphere Portal requires the LDAP server to be available at all times to be fully functional. In this section, we discuss how to make the LDAP server highly available.

In Chapter 4, “Securing WebSphere Portal with LDAP” on page 109, we set up the LDAP server to start only on the primary node. For a high availability scenario, we set up the LDAP z/OS server so that it can be started in what is called “multi-server” mode. This is a unique feature of LDAP on z/OS which is made possible by the fact that DB2 on z/OS provides database sharing across the sysplex.

When LDAP for z/OS is configured in multi-server mode, an instance of an LDAP server runs on each LPAR in the Parallel Sysplex, and each server shares the same TDBM database in DB2 by virtue of DB2 data sharing. We strongly recommend that you read Chapter 8 in *z/OS LDAP Server Administration and Use*, SC24-5923, before configuring multi-server mode.

You can configure LDAP multi-server mode in two ways:

1. So there is a failover relationship. In this scenario, one server is “active” and another in stand-by in case the primary LDAP fails.
2. So that all LDAP servers are active and there is dynamic workload balancing of requests by Sysplex Distributor.

To enable multi-server mode without dynamic workload balancing, the LDAP servers bind to a DVIPA but the LDAP ports are not distributed. In this case you must *omit* the sysplexGroupName and sysplexServerName properties in the

SLAPDCNF. If either of these keywords are present, they cause the LDAP server to operate in multi-server mode with dynamic workload management enabled.

To enable multi-server mode with dynamic workload balancing, the LDAP servers bind to a DVIPA and their port(s) are distributed by Sysplex Distributor. In this case you must also code the sysplexGroupName and sysplexServerName properties in the SLAPDCNF.

Because the sysplexServerName property must be unique for each LDAP server instance, this implies that you need a separate SLAPDCNF file for each LDAP server instance. There are also other considerations that have to be taken into account when Portal is accessing LDAP configured in this way. These considerations are described in 6.3.2, “Portal considerations with LDAP in multi-server mode” on page 219.

6.3.1 Enabling LDAP multi-server mode

Follow the steps provided in this section to alter the LDAP server installed in Chapter 4, “Securing WebSphere Portal with LDAP” on page 109 so that it runs in multi-server mode.

> **Note:** Our LDAP server is loading its configuration from a z/OS partitioned data set, HLQ.CNFOUT.
>
> However, remember that you can also configure it so the configuration files are held in the UNIX® HFS.
>
> In this section, we continue to use the partitioned dataset HLQ.CNFOUT.

Multi-server mode without dynamic workload balancing

In this scenario you do not code sysplexServerName, which is the only property that would force you to have separate SLAPDCNF members for each LDAP server instance. Therefore, you can share the existing SLAPDCNF between the LDAP server instance. In order to enable multi-server mode, make the following changes to the SLAPDCNF:

1. Set the listen property so the domain name is a DVIPA name.
2. Add the multiserver y property to the TDBM section.

Multi-server mode with dynamic workload balancing

In this scenario you must code sysplexGroupName in the SLAPDCNF. The sysplexServerName property must be unique for each LDAP server instance, which means that you must have separate SLAPDCNF members for each LDAP server instance. In order to enable multi-server mode in this scenario, make the following changes:

1. Copy the existing SLAPDCNF member in HLQ.CNFOUT and create a new member for each LDAP server instance. The new member names should use the LPAR name (the name resolved by system symbol &SYSNAME) as part of the member name.

   For example, in a sysplex with LPARs SC53 and SC69, you might copy the SLAPDCNF to members called LDAPSC53 and LDAPSC69.

2. Change the LDAPSRV started task JCL in SYS1.PROCLIB so the //SLAPDCNF DD card points to HLQ.CNFOUT(LDAP&SYSNAME).

3. In each of the new LDAP&SYSNAME members, make the following changes:

   a. Set the listen property so the domain name is a DVIPA name.

   b. Add the multiserver y property to the TDBM section.

   c. Add the sysplexGroupName and sysplexServerName properties. The sysplexGroupName must be the same for each LDAP server instance sharing the same TDBM database in DB2. A good choice for a name might use the sysplex name or a simple string like LDAPSRV.

      The sysplexServerName must be unique for each LDAP server instance. A good choice is to use the member name of the LDAP server's SLAPDCNF file, for example, LDAPSC53.

6.3.2 Portal considerations with LDAP in multi-server mode

When the LDAP z/OS server is configured in multi-server mode with dynamic workload balancing, there are e additional considerations if the Portal is using that LDAP server.

The issue is that, when you run the job to enable security for Portal, it sets a property for the LDAP user registry that will reuse connections. This is to reduce the overhead of binding to LDAP for each request. The Sysplex Distributor, however, does not maintain affinity for requests from a “client” (in this case, the client is Portal), so a BIND request can go to one LDAP server instance and the subsequent search request can go to another server instance where it will fail because that server never received a BIND request.

Therefore, in order to run Portal with a LDAP z/OS server configured in multi-server mode with dynamic workload balancing, you need to make the changes described next.

If using LDAP user registry without realms

1. Open the WebSphere Administration console and navigate to **Security -> Global Security**.
2. Select the user registry, either LDAP or Custom registry if you have configured Portal security with realms.
3. Uncheck the box `Reuse connections`, click **OK** and then save the changes.
4. Update the wmm.xml as described in the following section.

If using LDAP with or without realms

1. Run job EJPSCHOU from the .CNTL library that you created when using the Portal ISPF dialog to enable Portal security. This checks out the WMM configuration files so you can change them.
2. Locate the wmm.xml file, which will have been placed in `<app_server_home>/profiles/default/config/wmm` by the EJPSCHOU job.
3. Edit wmm.xml (it is in ASCII), set the following property: dirContextTimeToLive="0" and then save the changes.
4. Run job EJPSCHIN from the .CNTL library that you created when using the Portal ISPF dialog to enable Portal security. This checks in the WMM configuration files with the Deployment Manager, and the DMGR will synchronize the changes file to the node(s).
5. Restart the cell.

To verify the results of these changes, make sure the LDAP server can be started on each LPAR and that connections can be made to LDAP while its running on each LPAR. Other methods can be used to make the LDAP started task available on multiple LPARs, but this method worked best in our environment.

6.4 Full HA setup with test scenarios

This chapter has mostly focused on the construction of a WebSphere Portal cluster for high availability purposes. Our environment does have several other features which gives us increased availability. Figure 6-43 on page 221 is a complete diagram of all aspects of our highly available environment.

For a highly detailed description of all the elements in a highly available WebSphere Application Server environment, refer to the IBM Redbooks publication *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850, which is available at the following site:

`http://www.redbooks.ibm.com/abstracts/sg246850.html`

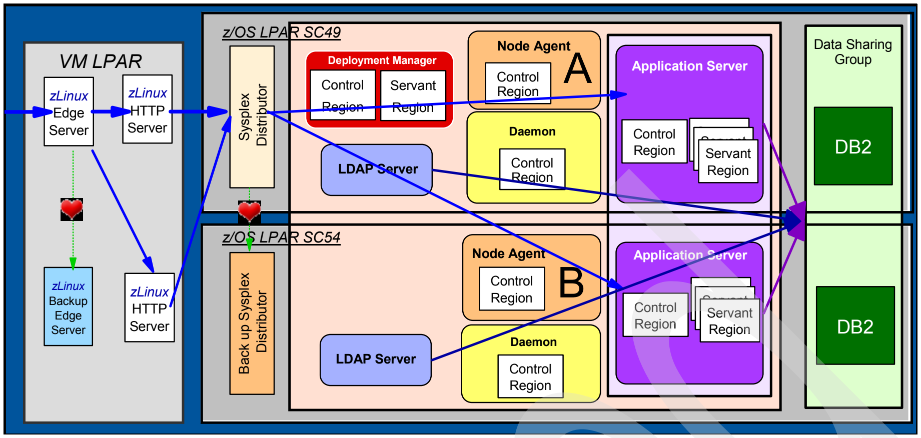


Figure 6-43 Complete HA WebSphere Portal environment

There are six components in total that create our highly available environment. Here is a description of each component and how the component is set up for high availability.

Edge server

When a request is first initialized, it comes into our environment through an edge server located on a z/Linux partition of the System z hardware. An *edge server* is a network dispatcher/load balancer which distributes work to our two HTTP severs. The edge server decides which HTTP server to send the request to based on the workload of each server.

The edge server has a backup which keeps in constant contact with the primary edge server to track its status. This monitoring of the second edge server is commonly known as a *heartbeat*. If the primary edge server goes down, the backup edge server takes over.

HTTP server

The HTTP server next sends the request to the Sysplex Distributor. The HTTP server is aware of all the application ports used by WebSphere Portal via the plugin-cfg.xml. For this reason, the end user does not need to enter a port in the WebSphere Portal URL. Our HTTP servers are highly available as well.

There are two HTTP servers defined to the environment. If one server goes down, the edge server sends all subsequent requests to the active HTTP server.

Sysplex Distributor

TCP/IP is installed in our environment with Sysplex Distributor enabled. Sysplex Distributor takes the request from the HTTP server and sends it out to the application servers.

It chooses an application server to distribute the workload over the two application servers based on information from WLM. If one of the application servers is down, the Sysplex Distributor will recognize this and send all requests to the available application server. The Sysplex Distributor is also highly available.

The backup Sysplex Distributor in the diagram always has a heartbeat connection with primary. If the backup loses the heartbeat from the primary server, it knows the primary server is down and it takes over as the primary.

Application server
Creating a WebSphere Portal application server cluster is discussed thoroughly throughout this chapter. WebSphere Portal cannot function without its LDAP server and its DB2 data. It is important to make these two resources, which are heavily used by WebSphere Portal, highly available.

LDAP server
The LDAP servers control the IDs that are used to access WebSphere Application Server and WebSphere Portal. 6.3, “LDAP high availability” on page 217 describes setting up the LDAP server for high availability. In our case, the LDAP server can be started on either of our LPARs if one of the LPARs in unavailable.

DB2 datasharing
DB2 is used heavily in a WebSphere Portal environment to store back-end data for the product. It is also used to store user ID information used by the LDAP server. Our DB2 subsystems are made highly available through DB2 datasharing.

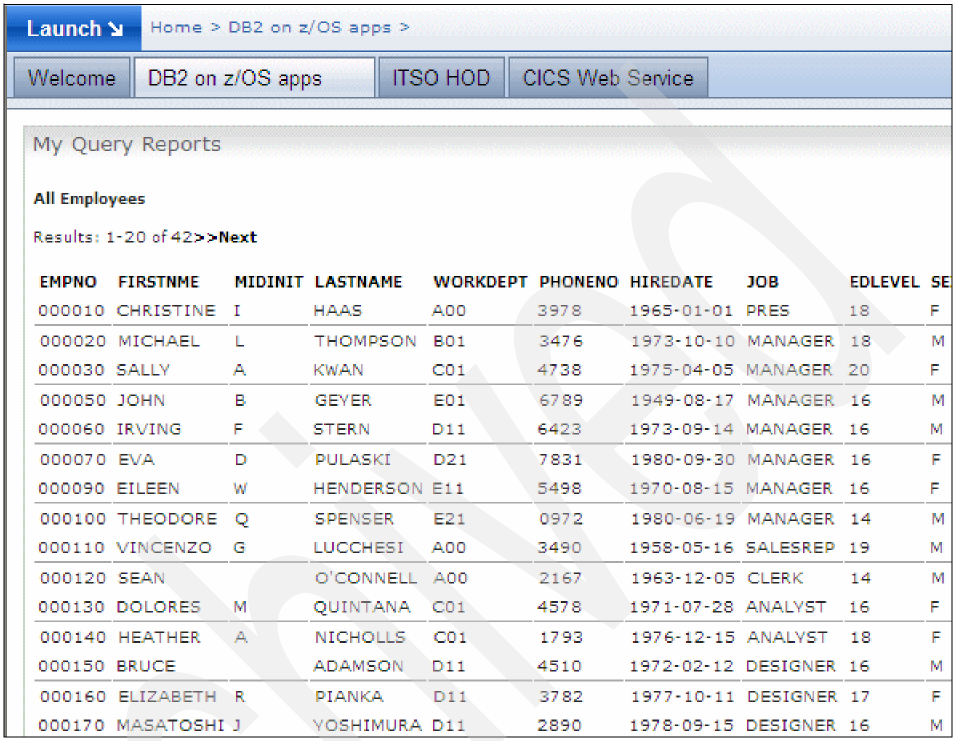
6.4.1 HA test scenario

We wanted to make sure that the clustered WebSphere Portal environment actually redirected work in the case of an application server failure. To accomplish this goal, a script was developed in WebSphere Studio Workload Simulator. For more information about this topic, refer to the following site:

http://www.ibm.com/software/awdtools/tester/performance/zos/index.html

WebSphere Studio Workload Simulator scripts record the actions of a user in an application environment. The script we recorded logs in to WebSphere Portal and selects a tab which has a custom portlet installed on it. When it gets to the custom portlet, it runs a query which selects all employees from a DB2 table. The result set spans three pages, so the script scrolls through the results by clicking Next twice. Finally, the script logs out of WebSphere Portal. An example of the custom portlet is shown in Figure 6-44 on page 223.

After recording the actions for the script, the script was enhanced to log the jsession ID cookie, the clone ID of the Portal application server processing requests and other application-related messages.



All Employees

Results: 1-20 of 42>>**Next**

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SE
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18	F
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M
000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M

Figure 6-44 Custom portlet used for cluster testing

WebSphere Studio Workload Simulator also allows a predefined workload to be sent into an application environment. We used this product to send 10 users into WebSphere Portal and run through the script. After a user finishes the script, the user goes back to the beginning of the script so we always have 10 users in the Portal at any given time.

In Example 6-5, both application servers are started. The example shows how the cluster assigns the user to either WebSphere_Portala or WebSphere_Portalb to distribute the workload evenly.

Example 6-5 Script execution with both application servers running

```
04/04/2007    13:10:57    0007:============ Starting Portal DB2 Session ============
04/04/2007    13:11:03    0001: WAS Server WebSphere_Portala is running session
GMevjs2hQLFsXAvOD8BEVX1
04/04/2007    13:11:04    0005: WAS Server WebSphere_Portalb is running session
-E69uGDTPAO_J3oEcG4trzy
```

```
04/04/2007    13:11:05     0009: WAS Server WebSphere_Portalb is running session 
twDy_TMm5-5ui0ZQFIgQllz
04/04/2007    13:11:05     0009:============= Ending Portal DB2 Session =============
04/04/2007    13:11:05     0009:============= Starting Portal DB2 Session =============
04/04/2007    13:11:14     0003: WAS Server WebSphere_Portala is running session 
lOUoWavmvFPSAKUSNYyEJXB
04/04/2007    13:11:14     0003:============= Ending Portal DB2 Session ============= ]
```

We continued to let the script run for a minute and then killed the WebSphere_Portalb application server to simulate an application server crash. After WebSphere_Portalb was killed, we received the message shown in Example 6-6 in the WebSphere Studio Workload Simulator logs. This message tells us that the work in progress on WebSphere_Portalb at the time of the crash was moved over to the WebSphere_Portala application server.

Switching the work over to the WebSphere_Portala application server is transparent to the user. The same results occur if the application server is brought down cleanly via the WebSphere administrative console. This situation may occur if maintenance is being applied to one application server at a time to keep the other available.

Example 6-6 Failover example

```
04/04/2007    13:11:49     0009: >>> Session HTFCQFkd2Y2A6hKtytZOPY8 
was interrupted on Server WebSphere_Portalb, and restarted on server 
WebSphere_Portala
04/04/2007    13:11:49     0009: WAS Server WebSphere_Portala is 
running session HTFCQFkd2Y2A6hKtytZOPY8
```

7

Implementing integrated security

Implementing security requires having an administrator create and maintain many kinds of “security definitions”, such as user ID and group definitions, resource definitions, and rules which say who can access resources. There are several other aspects to the role of security administrator, but the majority of the work centers around these activities. In a large, modern enterprise the security administrator will also need to manage two important concerns: the need for a Single Sign-On (SSO) solution, and the need to be immediately responsive to new requirements.

Therefore, in order to work effectively the security administrator needs a single point of administration and control for all these aspects of that role. Centralized management of user IDs and passwords, and of certificates related to authentication, is a major requirement. Centralized management of authorization rules is also important. For example, the security administrator will need to define who has access to Portal and non-Portal resources (such as J2EE and legacy applications), define or change roles statically or dynamically, and configure Single Sign-On.

These requirements justify the need to integrate Portal for z/OS with an external security manager that can manage more than just z/OS resources.

As documented here, we used Tivoli Access Manager (TAM) as our external security manager, and we implemented a number of scenarios in the area of authentication and authorization.

In this chapter we introduce you to Tivoli Access Manager and to the architectural considerations to keep in mind when integrating WebSphere Portal on z/OS and Tivoli Access Manager. Then we describe the numerous implementation scenarios that we executed. Those scenarios cover both authentication and authorization.

The chapter includes the following topics:

- Overview of Web security with Tivoli Access Manage
- Overview of TAM security solution components
- Overview of TAM external authentication and authorization
- Architecting our TAM and Portal on z/OS solution
- Configuring our TAM and Portal on z/OS

7.1 Introduction to Web security with Tivoli Access Manager

This section discusses the concept of Web security, and introduces IBM Tivoli Access Manager.

Note: In the following sections we provide information about Web security and the role of Tivoli Access Manager at a functional level. However, we do not address the physical infrastructure. The diagrams we provide here can be implemented in different ways. For example, the LDAP can be implemented on a distributed system (UNIX, Windows and so on) or on z/OS.

7.1.1 Web security

Web security deals with applying and enforcing security objectives to Web technologies and infrastructures. In today's world, Web technologies and infrastructures not only serve static and dynamic contents to Internet and intranet users, but also participate in Service Oriented Architectures (SOA) by supporting Web Services. The more integrated information systems are, the more important it is to meet security objectives which include identification, authentication, authorization, integrity, confidentiality, auditing and non repudiation.

As with security in general, Web security has evolved to embrace modern concepts such as "security as a service", "creating trust between systems", and "federating security islands". For example, in this section we highlight the authentication service and the authorization service. Later we explain how trust is established between the security proxy and Portal.

In a typical Web infrastructure, as illustrated in a simplified form in Figure 7-1 on page 228, *network layer security* is enforced using two layers of firewalls creating a protected zone known as a demilitarized zone (DMZ).

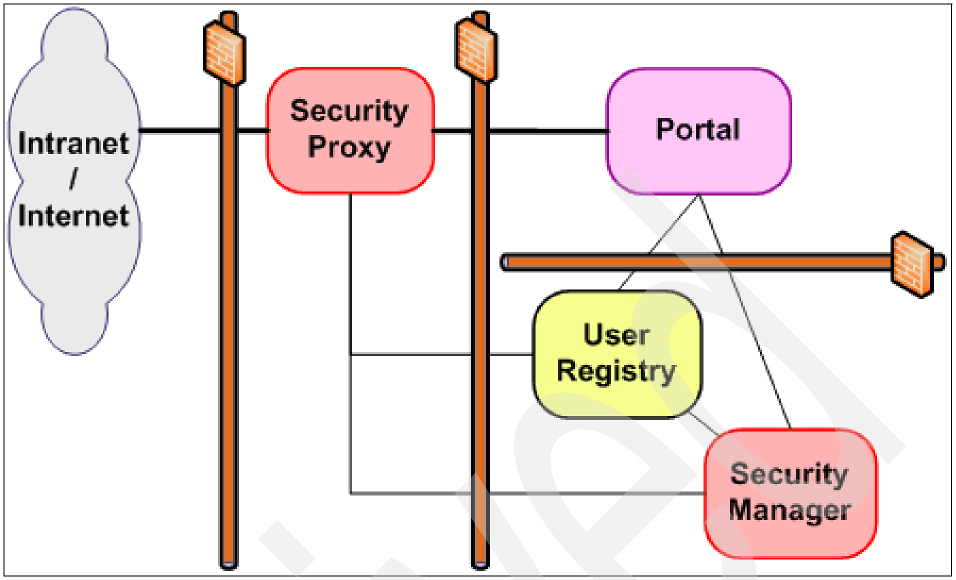


Figure 7-1 Generic Web security architecture

The DMZ provides a buffer between the external, untrusted Internet or intranet networks and a trusted internal production network. The DMZ concept enforces the defense in-depth principle of network design, which adopts an "onion-skin" approach.

Each layer of the onion is analogous to a network zone trust level. The more sensitive the data and applications, the closer to the center of the onion they should be deployed, thus providing layers of protection from less trusted networks. It is a best practice to separate, with a firewall, the secured production zone where Web applications are executed from the even more secured management zone where core security components are.

In a typical Web infrastructure, *application layer security* is enforced using components which provide at least authentication services and authorization services. It is a best practice to place the authentication service in the DMZ so that end users are filtered and known as early as possible.

The *authentication service* is commonly provided by a Reverse Proxy Security Server (RPSS) in the DMZ with assistance from a User Registry for identities and a Security Manager for policies in the management zone.

The *authorization service* can be provided by a Security Manager which possesses Access Control Lists (ACL) and policies. The Security Manager is responsible for the authorization decision-making process that helps to enforce security policies. Authorization decisions made by the authorization service result in the approval or denial of client requests to perform operations on protected resources.

There are advantages in externalizing the authentication and authorization services. For example, this approach centralizes the security management across heterogeneous environments. It also increases security by providing consistent and homogeneous role-based and policy-based security management.

These main authentication and authorization services are generally accompanied by other services to serve other security objectives. For example, security products widely use SSL to meet integrity and confidentiality security objectives. These security products usually also provide logging features to satisfy the auditing security objective.

In this chapter, we demonstrate how IBM Tivoli Access Manager provides Web security services to WebSphere Portal on z/OS.

7.1.2 IBM Tivoli Access Manager

IBM Tivoli Access Manager for e-business is a policy-based access control solution for enterprise applications such as WebSphere Portal on z/OS. It can help you to manage growth and complexity, control escalating management costs, and address the difficulties of implementing security policies across a wide range of Web and application resources.

Tivoli Access Manager for e-business integrates with Web applications right “out of the box” to deliver a secure, unified and personalized Web experience. Web-based Single Sign-On (SSO) can span multiple sites or domains by exploiting Tivoli Access Manager cross-domain SSO technology, or by using Security Assurance Markup Language (SAML) and other token-passing protocols.

Tivoli Access Manager for e-business allows you to define a comprehensive policy and administer security based on that policy, whether it is based on user roles or business rules. It provides security to heterogeneous environments.

With this architecture in place, access control is based on a single, consistent layer that allows for applications to be deployed faster and for security to be more accurately and consistently managed than in the “islands of security” approach.

The Tivoli Access Manager security solution provides and supports the following core technologies:

- Authentication

  Authentication is the first step a user must take when making a request for a resource that is protected by Tivoli Access Manager. During authentication, a user's identity is validated. Tivoli Access Manager allows a highly flexible approach to authentication through the use of frameworks and interfaces.

- Authorization

  Authorization enforces the security policy by determining the objects a user can access and the actions a user can take on those objects, and then granting appropriate access to the user. Tivoli Access Manager handles authorization through the use of the following:

  - Tivoli Access Manager authorization service
  - Access Control Lists (ACLs), Protected Object Policies (POPs), and authorization rules for fine-grained access control
  - Standards-based authorization API, using the aznAPI for C language applications, and the Java Authentication and Authorization Service (JAAS) for Java language applications
  - External authorization service capability

- Quality of protection

  The quality of protection is the degree to which Tivoli Access Manager protects any information transmitted between client and server. The quality of data protection is determined by the combined effect of encryption standards and modification-detection algorithms.

  The resource manager is responsible for ensuring that the quality of data protection is enforced. Quality of protection levels include standard Transmission Control Protocol (TCP) communication (no protection) and data integrity and data privacy provided by the Secure Sockets Layer (SSL) communication protocol.

- Scalability

  Scalability is the ability to respond to increasing numbers of users who access resources in the domain. Tivoli Access Manager uses the following techniques to provide scalability: replication of services; front-end replicated servers; back-end replicated servers; optimized performance by allowing the of-loading of authentication and authorization services to separate servers; scaled deployment of services.

- Accountability

  Tivoli Access Manager provides a number of logging and auditing capabilities. Log files capture any error and warning messages generated by Tivoli Access Manager servers. Audit trail files monitor server activity.

- Centralized management

  Three methods are provided for managing security policy and the Tivoli Access Manager servers:

  – The pdadmin command line utility

  – Web Portal Manager Graphical User Interface (GUI)

  – The administration API

You can accomplish most tasks using any of these methods. However, some tasks cannot be performed by using the Web Portal Manager.

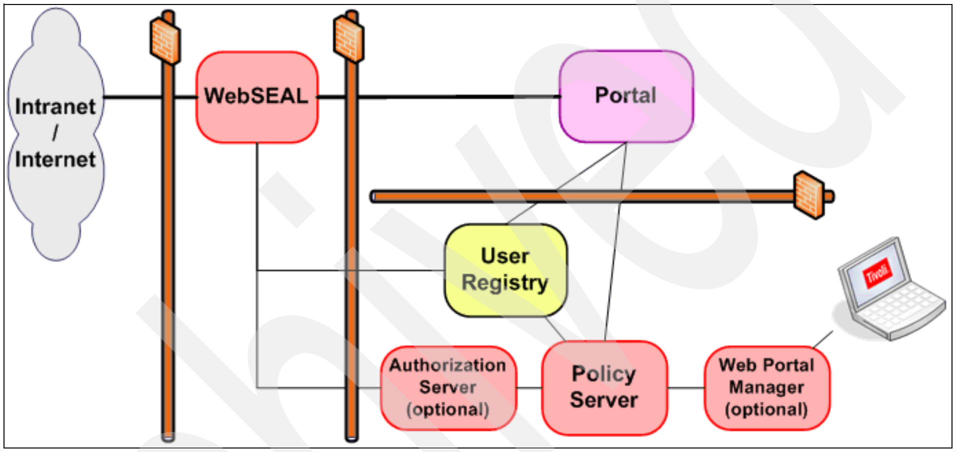


Figure 7-2 IBM Tivoli Access Manager typical components

An IBM Tivoli Access Manager for e-business solution is usually composed of the following components, as shown in Figure 7-2:

- User registry

  The user registry supports the Access Manager authorization functions. The registry provides a database of the user identities known to IBM Tivoli Access Manager. It also provides a representation of groups in IBM Tivoli Access Manager roles that may be associated with users. Finally, it provides a data store of metadata required to support additional functions.

- Policy Server

  The Policy Server maintains the master authorization database for the secure domain. This server is key to the processing of access control, authentication, and authorization requests. It also updates authorization database replicas by using push and pull methods, and it and maintains location information about other IBM Tivoli Access Manager servers in the secure domain.

Note that there can be only one instance of the Policy Server and its master authorization database in any secure domain at one time. For availability purposes, a standby server can be configured to take over policy server functions in the event of a system failure.

- WebSEAL

  WebSEAL is a high-performance, multi-threaded reverse proxy that sits in front of back-end Web applications. It applies a security policy to a protected object space. WebSEAL can provide Single Sign-On (SSO) solutions, and incorporate back-end Web application server resources into its security policy.

  Because it is implemented on an HTTP server foundation, WebSEAL is limited to enforcing policy for applications communicating with HTTP and HTTPS protocols. It uses junctions to define the connectivity to back-end servers.

- Authorization Server (optional)

  The Authorization Server may be installed to offload authorization decisions from the Policy Server, and provide for higher availability of authorization functions. The Policy Server provides updates for authorization database replicas maintained on each Authorization Server. The Authorization Server is an optional component.

- Web Portal Manager (optional)

  The Web Portal Manager is a Web-based graphical user interface (GUI) used for IBM Tivoli Access Manager administration. Similar to the pdadmin command line interface, this GUI provides management of users, groups, roles, permissions, policies, and other IBM Tivoli Access Manager tasks.

  A key advantage of the Web Portal Manager over the pdadmin command line utility is the fact that it is a browser-based application that can be accessed without installing any Access Manager-specific client components on the administrator’s local machine or requiring special network configuration to permit remote administrator access.

Instead of using a Reverse Proxy Security Server (RPSS) such as WebSEAL, you might choose to use a Tivoli Access Manager plug-in running in a Web server (Plug-in for Web Servers) or running in an Edge server (Plug-in for Edge server).

New with Version 6 of Tivoli Access Manager, the optional Policy Proxy Server enables Tivoli Access Manager applications and authorization servers to connect to a Policy Proxy Server rather than the Policy Server itself. The addition of a separate physical machine running Policy Proxy Server enables an architecture to be created where the only incoming SSL sessions to the Policy Server come from the Policy Proxy Server.

This facilitates increased security because a firewall protecting the Policy Server only has to allow inbound connections from the Policy Proxy Servers rather than from all Tivoli Access Manager applications or authorization servers.

Also new with Version 6 of Tivoli Access Manager, the optional Session Management Server (SMS) manages user sessions across complex clusters of Tivoli Access Manager security servers, ensuring that session policy remains consistent across the participating servers.

Using the Session Management Server allows WebSEAL and Plug-in for Web Servers to share a unified view of all current sessions and permits an authorized user to monitor and administer user sessions. The Session Management Server permits the sharing of session information, makes session statistics available, and provides secure and high-performance failover and single sign-on capabilities for clustered environments.

7.1.3 External authentication with Tivoli Access Manager

IBM Tivoli Access Manager provides an *external authentication service* with its WebSEAL component. WebSEAL is a Reverse Proxy Security Server (RPSS).

WebSEAL is placed in the DMZ. It receives all end-user requests. It provides the authentication service (with the help of the User Registry) to validate identities and passwords, and (with the help of the Policy Server) to conform to defined security policies. Then WebSEAL, based on junctions, establishes trust relationships with back-end servers and forwards requests along with end-user identities. The process of transferring the end-user identity from WebSEAL to WebSphere Portal is called *identity propagation* or Single Sign-On (SSO).

When requests reach Portal, Portal validates the trust with WebSEAL and validates the propagated identity using the User Registry. If the validation is successful, the Portal authorizes the request by checking whether the end user has the required permissions to access the Web resource. If so, the Web resource is delivered back to the WebSEAL server, which then gives the resource to the end-user.

Figure 7-3 on page 234 illustrates external authentication with Tivoli and Portal products.

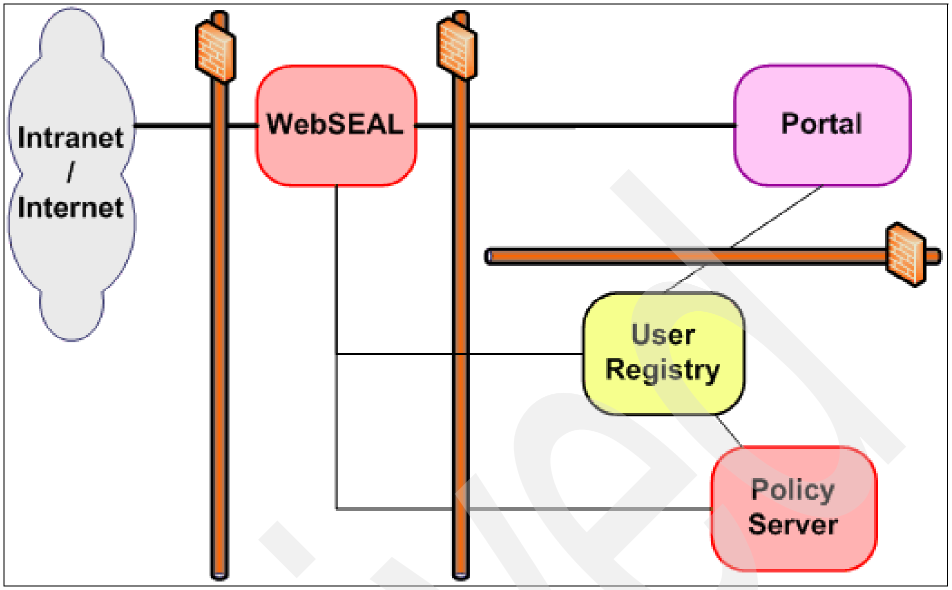


Figure 7-3 External Authentication with Tivoli and Portal products

The back-end servers to which WebSEAL can proxy are defined via *junctions*, which define a set of one or more back-end Web application servers that are associated with a particular URL. Traditional WebSEAL junctions are created by defining a new point in the URI space that indicates to WebSEAL which server to direct the request to.

Virtual Host junctions preserve the traditional Web addresses that may already exist within a corporation. *Transparent path junctions* remove the need for the junction name (such as /content/xyz) to be included in the Web address. Instead, transparent path junctions are part of the existing URI space located on the back-end server.

An important factor for a centralized security Portal solution is *trust*. If you configure all information requests to be routed through a central WebSEAL reverse proxy, you only want to authenticate the user once. This approach implies that all back-end application servers trust all incoming user requests as being properly authenticated by a preliminary authority such as WebSEAL.

WebSEAL can establish trust with WebSphere Portal relying on different mechanisms such as sharing keys (for example, Lightweight Third-Party Authentication (LTPA) keys), or such as providing a secret password (with the Trust Association Interceptor (TAI), for example).

After trust is established between WebSEAL and WebSphere Portal, WebSphere Portal can receive the end-user identity using identity propagation or Single Sign-On (SSO).

Single Sign-On is a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where that user has access permission, without the need to enter multiple passwords. Thus Single Sign-On includes the process of forwarding information about a user's identity in a secure way to another system.

WebSEAL supports several mechanisms for identity propagation including:

- Providing the user's identity via HTTP header values, which can be read and interpreted by the back-end server.
- Insertion of an HTTP Basic Authentication (BA) header to provide the back-end server with login information for the user, including a password. Optionally, this Basic Authentication header can permit login to the back-end server with a different identity from the one for the user who is logged in to WebSEAL.
- For junctions that support it (for example, with WebSphere Portal on z/OS), insert a Lightweight Third-Party Authentication (LTPA) token identifying the user into the HTTP stream that is passed to the back-end server.
- For junctions that support it (for example, with WebSphere Portal on z/OS), the use of a Trust Association Interceptor Plus (TAI or TAI++) to forward IBM Tivoli Access Manager credential information and establish trust between WebSEAL and back-end application server.

7.1.4 External authorization with Tivoli Access Manager

IBM Tivoli Access Manager provides an external authorization service with its Policy Server component. Optionally, the Authorization Server component can be used to offload some authorization workload from the Policy Server.

When an authenticated end user tries to access a protected resource, WebSphere Portal asks the Policy Server for authorization. The Policy Server executes the decision-making process based on Access Control Lists (ACL), on Protected Objects Policies (POP) and also on authorization rules. After making the decision, the result of the decision is sent back to Portal. Portal then executes the decision by giving or refusing access to the protected resource.

Figure 7-4 on page 236 illustrates external authorization architecture with Tivoli and Portal products.

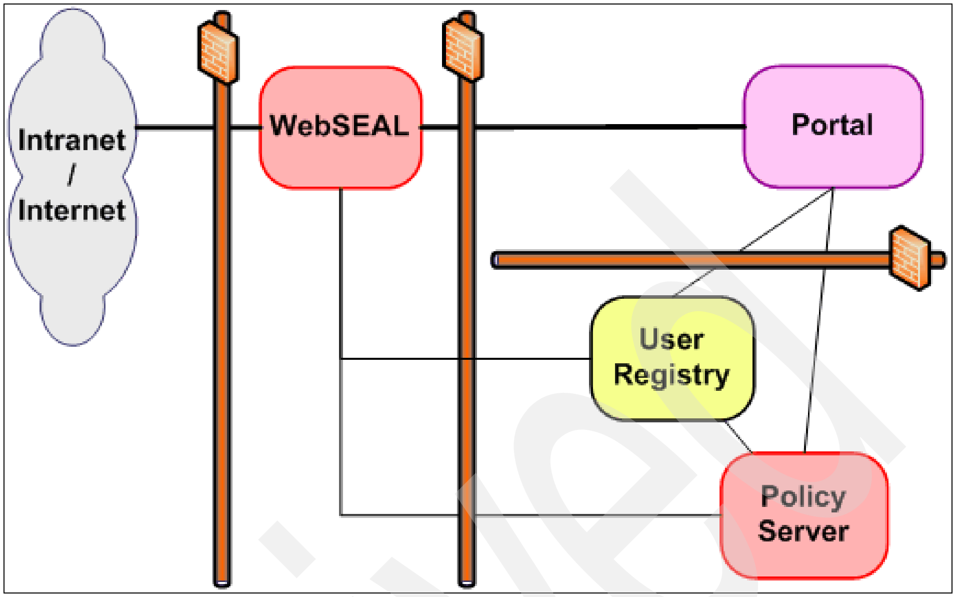


Figure 7-4 External authorization architecture with Tivoli and Portal products

Note: With WebSphere Portal on z/OS, if you use IBM Tivoli Access Manager for authorization, you must also use it for authentication.

Using Tivoli Access Manager to perform only authorization is *not* supported.

With IBM Tivoli Access Manager, the security policy is defined using a combination of:

- Access Control Lists (ACLs)

  An Access Control List (ACL) specifies the predefined actions that a set of users and groups can perform on an object. For example, a specific set of groups or users can be granted read access to the object.

- Protected Object Policies (POPs)

  A Protected Object Policy (POP) specifies access conditions associated with an object that affects all users and groups. For example, a time-of-day restriction can be placed on the object that excludes all users and groups from accessing the object during the specified time.

- Authorization rules

  An authorization rule specifies a complex condition that is evaluated to determine whether access will be permitted. The data used to make this decision can be based on the context of the request, the current environment, or other external factors. For example, a request to modify an object more than five times in an 8-hour period could be denied.

A security policy is implemented by strategically applying ACLs, POPs, and authorization rules to those resources requiring protection. The IBM Tivoli Access Manager authorization service makes decisions to permit or deny access to resources based on the credentials of the user making the request and the specific permissions and conditions set in the ACLs, POPs, and authorization rules.

7.2 Architecting our TAM and Portal on z/OS solution

This section describes the architecture of our environment and highlights some security integration scenarios.

7.2.1 Our TAM and Portal on z/OS environment

In this section, we explain and illustrate how to integrate WebSphere Portal on z/OS with an external authentication service and with an external authorization service. For this purpose we leverage robust IBM Tivoli Access Manager infrastructure capabilities such as centralized security management, fine-grained security, and auditing.

For the *external authentication service*, different hardware and software solutions exist. It is possible to use a hardware appliance such as IBM WebSphere Datapower XS40, which is a network device that can integrate with IBM Tivoli Access Manager. It is also possible to use TAM Plug-in for Edge Server or TAM Plug-in for Web Servers. These plug-ins integrate with TAM and enforce authentication respectively at the WebSphere Edge Server level or Web server level.

Alternatively, it is possible to use the TAM WebSEAL software solution. In our environment we choose WebSEAL because it is very flexible and contains advanced configuration features. Moreover, some WebSEAL and WebSphere Portal on z/OS security configuration are available “out of the box” because there are ISPF dialogs to set up the integration.

For the *external authorization service* in our environment, we choose to use the TAM Policy Server along with the Authorization Server. IBM Tivoli Access Manager integrates with WebSphere Portal on z/OS using the AMWAS module running in WebSphere Portal.

The Java Authorization Contract for Containers (JACC) standard is not used between TAM and WebSphere Portal because JACC is only supported in WebSphere V6.1 onward, and the AMWAS module provides greater flexibility and better granularity.

> **Important:** Using an external security manager to perform only authorization is not supported at this time. In order to perform authorization with an external security manager, authentication using an external security manager needs to be set first.

It is a good choice to run IBM Tivoli Access Manager components on Linux for System z in order to consolidate servers with hardware resources virtualization and in order to benefit from HiperSockets™ communications between TAM components and WebSphere Portal on z/OS. HiperSockets provides in-memory TCP/IP connections between and among LPARs. They operate at memory speed for maximum performance.

HiperSockets also provides security benefits, especially on the memory key-protected mainframe because there is no opportunity to intercept a network connection. Moreover, HiperSockets improves reliability and availability because there are no network hubs, routers, adapters, or wires to break. In our environment we reused an existing TAM infrastructure running on MS Windows.

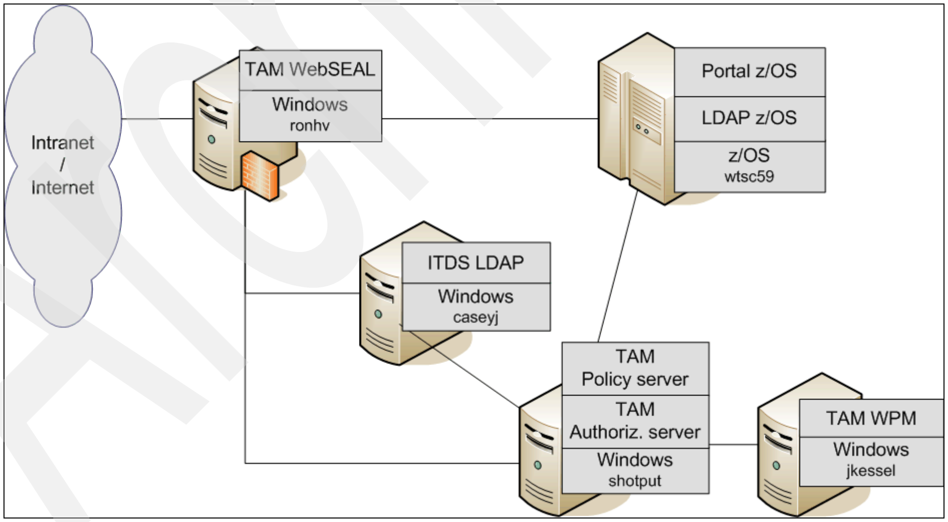


Figure 7-5 Our external authentication architecture

As shown in Figure 7-5 on page 238, our environment had two user registries. LDAP on z/OS was used by WebSphere Portal on z/OS. IBM Tivoli Directory Server (ITDS) was used by TAM on a Windows machine.

In our case, we used two LDAPs because the TAM infrastructure including an LDAP was already in place on the Windows server, and the Single Sign-On was being managed through this server. We established the LDAP server on z/OS as part of the WebSphere Portal configuration process. We found this situation to be acceptable as long as we could synchronize the LDAP identities and groups and ensure they had the same distinguished names on both LDAP servers.

It is common to synchronize registries so that only core pertinent information is shared across the enterprise. Some products store in LDAP servers some information that is product-specific and which is not necessary for other pieces of the enterprise infrastructure. Synchronization allows choosing what LDAP information to share across the enterprise. WebSphere Portal on z/OS has a feature to enable automatic user provisioning to TAM. Some advanced solutions based on IBM Tivoli Directory Integrator (ITDI) can also be used.

> **Note:** Each time a new solution or product using LDAP has to be installed on z/OS, you need to make two decisions:
>
> - Will a new LDAP be set up and used specifically for the product or solution, or will an existing LDAP be used?
> - Is there a preference for a z/OS LDAP or a remote LDAP?
>
> In any case, when multiple LDAPs are in use and Single Sign-On has to be achieved, there must be a mechanism to synchronize information between the LDAPs.

It is a best practice to deploy IBM Tivoli Access Manager components (WebSEAL, ITDS, Policy Server, Web Portal Manager) on separate machines or LPARs for reliability and flexibility reasons.

Our environment had only one Authorization Server embedded with the Policy Server because we did not need a separate standalone TAM Authorization Server. Standalone Authorization Servers are mainly used for scalability purposes in order to offload work from the main Policy Server. In our scenario, we did not test scalability because the focus was on integrating TAM with WebSphere Portal on z/OS. Our environment also did not need a TAM Policy Proxy Server because we did not figure network security with firewalls.

WebSEAL communicates with IBM Tivoli Directory Server (ITDS) for userid and password authentication, and with the Policy Server for enforcing security policies and some authorization. WebSEAL communicates with WebSphere

Portal on z/OS to forward authenticated requests from end users to WebSphere Portal. The Policy Server communicates with ITDS for user and group information, and for storing some configuration information. WebSphere Portal on z/OS communicates with the Policy Server for authorization purposes. The Web Portal Manager (WPM) communicates with the Policy Server to administrate the TAM environment using a Web console.

As mentioned, the focus in our case was on security integration capabilities and not on scalability. For this reason, our environment had only one instance of each component. Nevertheless, all components involved in this architecture are designed for real-life production architectures and possess scalability features such as load balancing, affinity or sessions synchronization.

7.2.2 External authentication using TAI for SSO

In our environment, we choose TAM WebSEAL to provide the external authentication service. A common end-user requirement is identity propagation or Single Sign-On (SSO) between the Reverse Proxy Security Server (RPSS) and the Web application server. As described, WebSEAL can establish trust and propagate identity based on different mechanisms. In this section, we focus on the Trust Association Interceptor (TAI) mechanism.

WebSphere Portal relies on WebSphere Application Server capabilities for Single Sign-On. It supports Single Sign-On with external authentication services such as RPSS through “trust associations”. When trust associations are enabled, WebSphere Portal is not required to authenticate a user if a request arrives via a trusted source that has already performed authentication.

Trust association enables the integration of WebSphere Portal security and third-party security servers. More specifically, a RPSS can act as a front-end authentication server while WebSphere Portal applies its own authorization policy onto the resulting credentials that are passed by the RPSS. The idea that WebSphere Portal can support trust association implies that the product security recognizes and processes HTTP requests that are received from a RPSS.

WebSphere Portal and the RPSS engage in a contract in which the product gives its full trust to the RPSS, and the RPSS applies its authentication policies on every Web request that is dispatched to WebSphere Portal. This trust is validated by the interceptors that reside in the Portal environment for every request received. The method of validation is agreed upon by the proxy server and the interceptor.

The external authentication service is expected to:

- Establish trust with WebSphere Portal
- Perform user authentication
- Insert user credential information into HTTP requests

The module in WebSphere Portal which handles the trust association is the Trust Association Interceptor (TAI). It is a pluggable module whose responsibilities are:

- Validation of trust with the external authentication service
- Extraction of credential information from the request

TAM WebSEAL and WebSphere Portal integration is facilitated by the presence of two Tivoli TAIs provided with the WebSphere Portal product:

- WebSealTrustAssociationInterceptor supports identity propagation only and returns a String to WebSphere Portal.
- TAMTrustAssociationInterceptorPlus (which is newer) supports attribute propagation (identity and attributes) and returns a Subject to WebSphere Portal.

In the following sections, we explain these TAIs in more detail.

WebSealTrustAssociationInterceptor

The WebSealTrustAssociationInterceptor TAI is provided to support WebSEAL Version 4.1. If you plan to use WebSEAL Version 5.1 or higher, we recommend that you migrate to use the newer TAMTrustAssociationInterceptorPlus.

The WebSealTrustAssociationInterceptor TAI is configured by using the WebSphere Portal z/OS customization dialogs, as explained in 7.3.1, “External authentication using Tivoli Access Manager and the Trust Association Interceptor” on page 249. The WebSphere Portal on z/OS dialogs and jobs allow configuring the TAI running in WebSphere Portal on z/OS and also the associated WebSeal junction to forward proper requests. .

Using the WebSealTrustAssociationInterceptor, trust is established between WebSEAL and WebSphere Portal by authenticating a WebSEAL user ID and password against the user registry, which is LDAP on z/OS. Using the WebSealTrustAssociationInterceptor, the end-user identity is propagated from WebSEAL to WebSphere Portal using the specific *iv-user* HTTP header.

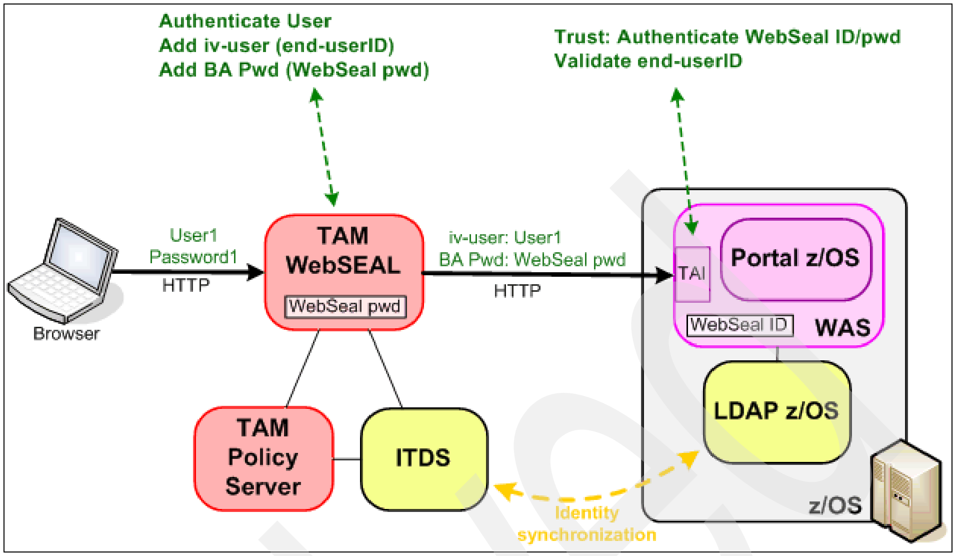


Figure 7-6 WebSphere Portal on z/OS external authentication using TAI for Single Sign-On

Figure 7-6 shows how this scenario works:

1. The end user authenticates to WebSEAL using one of the WebSEAL-supported authentication methods. The end user provides a user ID (User1) and a password (Password1).
2. WebSEAL authenticates the end user against its user registry, which is IBM Tivoli Directory Server (ITDS). It verifies that the user ID (User1) and password (Password1) are correct against the ITDS user registry.
3. If authentication is successful, WebSEAL forwards the HTTP request to WebSphere Portal including some additional information. For identity propagation, it adds an iv-user HTTP header which contains the end-user identity (User1). Note that only the user ID is transferred, not the LDAP distinguished name.

   For establishing the trust with WebSphere Portal, it adds a Basic Authentication Password (BA Pwd) HTTP header which contains the WebSEAL identity password. This password is configured in the WebSEAL configuration file. There is no end-user password transferred between WebSEAL and WebSphere Portal.
4. WebSphere Portal on z/OS receives the HTTP request from WebSEAL. The TAI intercepts the request and first validates the trust. For this purpose it extracts the WebSEAL identity password from the request Basic Authentication Password HTTP header. It reads the WebSEAL identity

user ID from the Portal TAI configuration and authenticates the WebSEAL identity user ID and password against the WebSphere Portal user registry, which is LDAP on z/OS. After the WebSEAL user ID and password are validated, the trust is established.

5. Then TAI retrieves the end-user identity. For this purpose it extracts the end-user identity from the request iv-user HTTP header and gives this identity userid as a String to WebSphere Portal. There is no end-user authentication at this point.

   Actually, end-user authentication would not be possible because WebSphere Portal does not know the end-user password. The TAI bypasses the normal authentication process. After the TAI is run, the end user appears as authenticated in WebSphere Portal.

TAMTrustAssociationInterceptorPlus

WebSealTrustAssociationInterceptor is able to provide only the user ID to the WebSphere Portal runtime. After this TAI is invoked, further user registry searches are required to create the various credentials required for authorization.

To support the return of *complete* credential information, the TAI interface has been enhanced with TAMTrustAssociationInterceptorPlus. This means that no additional user registry searches are required after the TAI invocation.

TAMTrustAssociationInterceptorPlus is not configured using the Portal z/OS customization dialogs. The WebSphere Portal on z/OS customization dialogs only allows configuring the WebSealTrustAssociationInterceptor.

In order to configure WebSphere Portal on z/OS with TAMTrustAssociationInterceptorPlus, you must manually configure the underlying WebSphere Application Server for z/OS TAI using the WebSphere administrative console or wsadmin. You also need to configure the WebSEAL junction so that it can establish the trust and also forward the iv-creds HTTP header.

A WebSEAL junction needs to be created between WebSEAL and WebSphere Application Server, thus ensuring that the iv-creds and the HTTP Basic Authentication headers are passed in the request. The Basic Authentication header should contain the password for the WebSEAL identity. The username in the Basic Authentication header is incidental and the value does not matter. The TAI will use the loginid property value configured in WebSphere as the user to authenticate with the password in the Basic Authentication header.

WebSphere Application Server calls a TAI method to establish trust with the external authentication server and retrieve the credentials. This method establishes trust with WebSEAL by checking that the BA header contains the correct password for the configured WebSEAL identity.

The TAM Authorization Server is contacted to make this decision. The iv-creds header is then extracted from the request and used to construct a PDPrincipal object. A credential object containing user and group information is constructed from information contained in the PDPrincipal.

The Principal and the Credential objects are inserted into a JAAS Subject, which is returned from the call. At this point WebSphere Application Server has valid credentials that it can use for making authorization decisions in the usual J2EE manner. In addition, the Subject now contains the PDPrincipal object, which application code can access if needed.

Note the following difference:

- With TAMTrustAssociationInterceptorPlus, WebSEAL will insert the iv-user header *and* the iv-creds header into the HTTP request.
- With WebSealTrustAssociationInterceptor, WebSEAL will insert *only* the iv-user header into the HTTP request.

The TAMTrustAssociationInterceptorPlus does not directly contact LDAP (unlike the WebSealTrustAssociationInterceptor). Instead, IT contacts a TAM Authorization Server which validates the WebSeal identity password to establish trust with WebSEAL. This means that additional configuration is required on the WebSphere Application Server side to ensure that the TAI can reach a TAM Authorization Server. An implication of this is that there has to be a “hole” through the firewall in order to allow WebSphere Portal to enter the Management Zone.

You can learn more details about the TAMTrustAssociationInterceptorPlus at the following site:

http://www.ibm.com/developerworks/tivoli/library/t-tamtai/

7.2.3 External authentication using LTPA for SSO

In our environment, we choose TAM WebSEAL to provide the external authentication service. A common end-user requirement is identity propagation or Single Sign-On (SSO) between the Reverse Proxy Security Server (RPSS) and the Web application server. As described earlier, WebSEAL can establish trust and propagate identity based on different mechanisms. In this section we focus on an alternative configuration that uses the Lightweight Third-Party Authentication (LTPA) mechanism.

WebSphere Portal relies on WebSphere Application Server capabilities for Single Sign-On. It supports SSO with external authentication services such as RPSS through LTPA. When LTPA SSO is enabled, WebSphere Portal is not required to authenticate a user if a request arrives via a trusted source that has already performed authentication.

LTPA is an IBM proprietary technology which serves different purposes such as identity propagation, attribute propagation, Single Sign-On, and reuse of already authenticated identities across IBM WebSphere Application Server-based products and across IBM Domino products.

TAM WebSEAL also supports LTPA and can establish trust and propagate identities to WebSphere Portal using an LTPA token. WebSEAL junctions can provide the LtpaToken, which is also called the *authentication token* or the *interoperability token*. The LtpaToken only possesses the end-user identity.

WebSEAL junctions can also provide the newer LtpaToken2, which is also called the Single Sign-On (SSO) token or the Web inbound security attribute propagation token. The LtpaToken2 possesses not only the end-user identity, but also additional information for attribute propagation.

LTPA tokens are encrypted and decrypted using a LTPA key. This key is generated on the WebSphere side and then copied over to all trusted parties, such as WebSEAL.

When a user makes a request for a WebSphere resource, the user must first authenticate to WebSEAL. After successful authentication, WebSEAL generates an LTPA cookie on behalf of the user. The LTPA cookie, which serves as an authentication token for WebSphere, contains the user identity, key and token data and expiration information. This information is encrypted using a password-protected secret key shared between WebSEAL and the WebSphere server.

WebSEAL inserts the cookie in the HTTP header of the request that is sent across the junction to WebSphere. The back-end WebSphere server receives the request, decrypts the cookie, and validates the user based on the identity information supplied in the cookie. This cookie never reaches the end-user browser. It is included in the communication between WebSEAL and WebSphere only. WebSEAL maintains state by mapping incoming user information (SSL session ID, BA user ID) to the LTPA token.

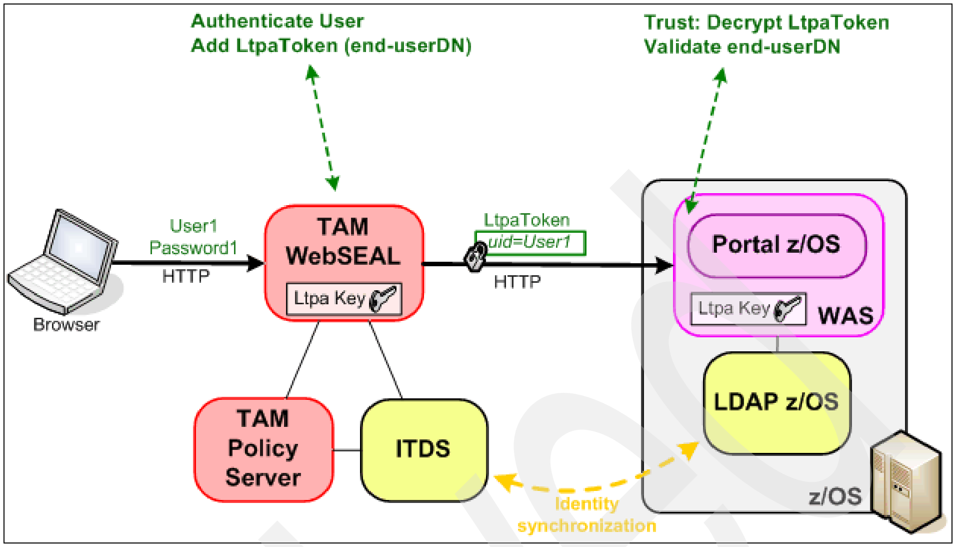


Figure 7-7 Portal z/OS external authentication using LTPA for Single Sign-On

Figure 7-7 illustrates how this scenario works:

1. The end user authenticates to WebSEAL using one of the WebSEAL supported authentication methods. The end user provides a user ID (User1) and a password (Password1).
2. WebSEAL authenticates the end user against its user registry, which is IBM Tivoli Directory Server (ITDS). It verifies that the userid (User1) and password (Password1) are correct against the ITDS user registry.
3. If authentication is successful, WebSEAL forwards the HTTP request and some additional information to WebSphere Portal. For identity propagation, it adds an LtpaToken cookie HTTP header which contains the end-user distinguished name (uid=User1). The complete LDAP distinguished name is transferred, not only the user ID.

   For establishing the trust with WebSphere Portal, it encrypts this LtpaToken with the Ltpa Key. There is no end-user password transferred between WebSEAL and WebSphere Portal.
4. WebSphere Portal on z/OS receives the HTTP request from WebSEAL. The Ltpa login module receives the request and first validates the trust. For this purpose it decrypts the LtpaToken with a local copy of the Ltpa Key.

   If decryption is successful, then it proves that the LtpaToken comes from a third party that possess the same Ltpa Key and which is consequently trusted. After the Ltpa token is decrypted, the trust is established.

Then the Ltpa login module reads the content of the Ltpa token and retrieves the end-user identity. The end-user LDAP distinguished name is validated against the user registry, which is LDAP on z/OS.

Note that there is no end-user authentication at this point. Actually end-user authentication would not be possible because Portal does not know the end-user password. After the Ltpa login module runs, the end user then appears as authenticated in Portal.

7.2.4 External authorization for WebSphere Portal on z/OS

WebSphere Portal on z/OS can rely on Tivoli Access Manager for resources access authorization. WebSphere Portal on z/OS can delegate this responsibility to Tivoli Access Manager, enabling it to provide centralized management of security policy both for WebSphere Portal resources and for resources unrelated to WebSphere Portal.

When Tivoli Access Manager is integrated with WebSphere Portal on z/OS in this way, Tivoli Access Manager determines whether a user has any of the roles necessary to access a requested Portal resource. The inputs are the same but the Portal role is now managed by Tivoli Access Manager. The authorization is stored in the Tivoli Access Manager ACL database.

Tivoli Access Manager supports JAASLogin for authentication (PDLogin java class) and PDPermission for authorization (PDPermission class). The PDLogin class knows how to authenticate to Tivoli Access Manager. The PDPermission class knows how to locate the current JAAS Subject, extract the authentication information, and contact Tivoli Access Manager to see if the current Subject is allowed the particular access to the particular resource.

Tivoli Access Manager integrates with WebSphere Portal on z/OS using the AMWAS module running in WebSphere Portal. Java Authorization Contract for Containers standard (JACC) is not used between TAM and Portal because the AMWAS module provides greater flexibility and better granularity.

With WebSphere Portal on z/OS, resource access authorization is managed by Tivoli Access Manager only if the WebSphere Portal on z/OS resource access is externalized. If the WebSphere Portal on z/OS resource access is internalized, WebSphere Portal on z/OS itself manages the resource access authorization. After a WebSphere Portal on z/OS role is externalized, Tivoli Access Manager is being used to add and remove users and groups to the Access Control List (ACL) for the role. You can use Tivoli Access Manager to provide access control for all public Portal resources, or for a subset of public Portal resources, depending on the needs of your environment. Access control for private pages cannot be externalized.

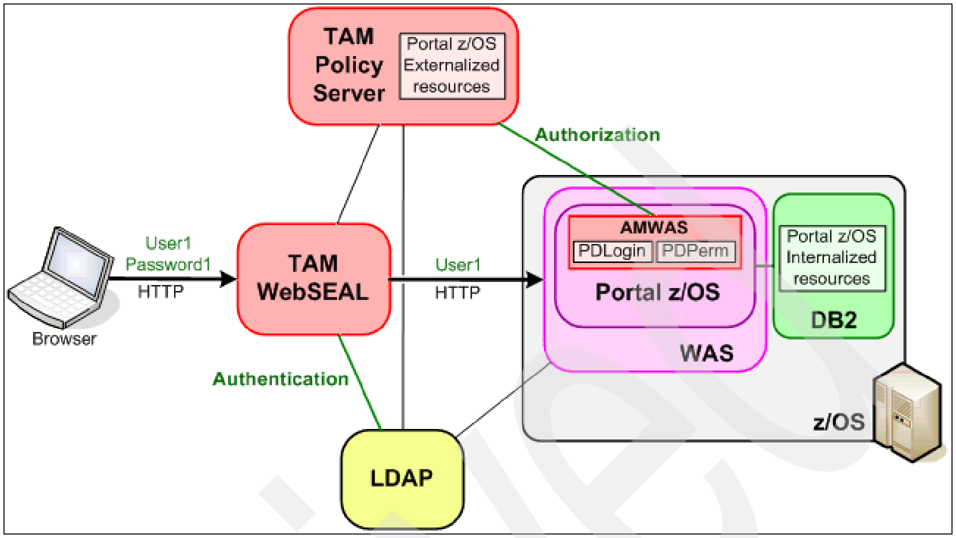


Figure 7-8 WebSphere Portal on z/OS external authorization with Tivoli Access Manager

Figure 7-8 illustrates an external authorization scenario with WebSphere Portal on z/OS. This scenario also includes external authentication with WebSEAL and Single Sign-On. Thus, the user ID (User1) of the end user is propagated to WebSphere Portal on z/OS.

For all WebSphere Portal on z/OS externalized resources, the authorization decision-making process is executed by Tivoli Access Manager.

When the request arrives in the Portal, the AMWAS PDLogin class is called to build TAM credentials along with the PDPrincipal object. This information is placed in the JAAS Subject for later authorization checks.

When the request tries to access externalized protected resources, the AMWAS PDPermission class is called to retrieve the end-user information from the JAAS Subject and to ask Tivoli Access Manager if the user is allowed to access the resource. Tivoli Access Manager analyses the ACL, the POP and the authorization rules in order to make the access or no-access decision. Then Tivoli Access Manager sends the decision back to Portal. WebSphere Portal on z/OS executes the decision and gives access (or refuses access) to the protected resource.

An implication of this is that a call must be made across the network from Portal to TAM for every authorization request that Portal has to make. Keep in mind that a single Portal page might contain several portlets, each with its own security rule, so choosing to use this configuration might have performance implications. Placing the TAM on z/Linux and using HiperSockets to communicate is a useful way to reduce this possible impact.

7.3 Configuring our TAM and Portal on z/OS

Because WebSphere Portal on z/OS runs as an application server in WebSphere Application Server, Portal inherits some of WebSphere Application Server's services such as security. Initially, authentication in Portal is handled by WebSphere Application Server and authorization is handled by WebSphere Portal.

In the implementation sections of this chapter we cover two scenarios:

- External authentication using the TAI and Tivoli Access Manager, discussed in 7.3.1, "External authentication using Tivoli Access Manager and the Trust Association Interceptor" on page 249
- External authorization using Tivoli Access Manager, discussed in 7.3.2, "External authorization using Tivoli Access Manager" on page 274

Note: In our description of the scenarios we assume that you have the required Tivoli Access Manager components already installed, available and accessible by the WebSphere Portal on z/OS. It is beyond the scope of this publication to describe the installation and customization of Tivoli Access Manager.

For information about installing Tivoli Access Manager, refer to *Distributed Security and High Availability with Tivoli Access Manager and WebSphere Application Server for z/OS*, SG24-6760.

7.3.1 External authentication using Tivoli Access Manager and the Trust Association Interceptor

When using an external security manager such as Tivoli Access Manager to handle authentication, a Trust Association Interceptor (TAI) is used by WebSphere Application Server to trust the external security proxy (in this case, WebSEAL).

When an external security manager is used, the process for authentication is as follows.

Whenever a request attempts to access a secured resource, WebSphere Application Server invokes the TAI, which validates that the request comes from a legitimate third-party authentication proxy and returns the user's authenticated identity to WebSphere Application Server. The TAI should return either a Distinguished Name (DN) or a short name.

WebSphere Application Server performs a registry lookup to verify the Distinguished Name or convert the short name to a Distinguished Name before searching for group memberships for that user. If the registry lookup fails, WebSphere Application Server rejects the request. If the registry lookup succeeds, WebSphere Application Server generates a Lightweight Third-Party Authentication (LTPA) token for the user and stores it as a cookie for subsequent authentication during the user's session.

TAIs are not necessary if the external security manager provides native support for WebSphere Application Server identity tokens, such as LTPA tokens.

In this implementation, we configure SSO for Portal using the “out-of-the-box” TAI provided by TAM using the Portal customization dialogs.

After the implementation we described here is complete, WebSphere Portal on z/OS users will access Portal via the newly created WebSEAL junction point. The Portal on z/OS users will authenticate with WebSEAL, bypassing the normal WebSphere Portal on z/OS login screen, and be directly presented with the Portal personalized page.

Configuring the TAI using the Portal customization dialog

In the following steps we use the Portal customization dialog to perform these tasks:

- Define variables to establish communications between WebSphere Portal on z/OS and Tivoli Access Manager (connection settings)
- Create a connection in TAM WebSEAL that will be recognized by WebSphere Application Server (WebSEAL junction settings)
- Enable authentication between WebSEAL and WebSphere Application Server (WebSEAL TAI settings)
- Establish SSL communication between the TAM Policy Server and WebSphere Portal on z/OS (Policy Server settings)

After the variables are defined, we generate a customization job for this task. The instruction job must be viewed before running any of the jobs.

Finally, we verify the external authentication configuration we tried to achieve.

> **Note:** WebSphere Portal security must be enabled before proceeding with this section.

The are the steps to follow:

1. On the ISFP Primary Option Menu, select option **6 - TSO or Workstation commands**.
2. Enter the command shown in Example 7-1 to execute the Portal's customization dialog, and press Enter to execute the command.

Example 7-1 Command to execute Portal's customization dialog

```
exec 'BBWP6050.SBBOCLIB(BBOWSTRT)' 'APPL(PS1) PROD(EJP) PRODHLQ(BBWP6050)'
```

WebSphere Portal is an add-on product to WebSphere Application Server. Access the Portal customization dialog through the WebSphere Application Server customization dialog as follows:

3. Select option **5 - WebSphere Application Server-based add-on products** and press Enter.

   > **Note:** Start the ISPF dialog using an APPL parameter that applies to the node where you configured security.
   >
   > In our case, we used APPL(PS1) because we had enabled security on the primary node using APPL(PS1).

   As more products are installed on WebSphere Application Server, this list will grow. You will see only one option at this point: `WebSphere Portal for z/OS`.

4. Select option **1 - WebSphere Portal for z/OS Configure WebSphere Portal** and press Enter.

   The next panel provides WebSphere Portal product information such as version.

5. Press Enter to clear the WebSphere Portal product information panel.

   This panel is the main configuration panel for customizing the Portal. Configuring an external security is a security configuration task.

6. Select option **3 - Security configuration tasks**.

   This panel provides a list of Security Configuration tasks. Configuring external authentication is under "Configure external security".

7. Select option **3 - Configure external security** and press Enter.

This panel provides tasks that WebSphere Portal on z/OS can use via an external security manger. Configuring WebSphere Portal on z/OS to use an external authentication server and a WebSphere Application Server TAI to provide Single Sign-On capabilities is a “Configure external authentication” task.

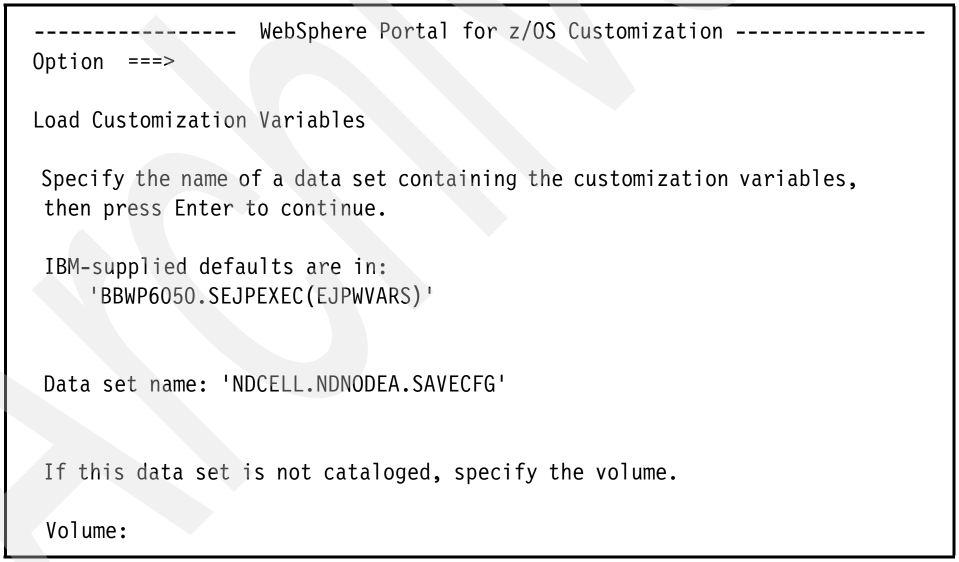
8. Select option **1 - Configure external authentication**.

   It is important to ensure that you are using the *correct* WebSphere Application Server configuration environment variables of the WebSphere Application Server that you will use for your Portal install.

   These WebSphere Application Server configuration variables get loaded into the system as the basis for the Portal configuration. In our case, we stored these variables in a data set called NDCELL.NDNODEA.SAVECFG.

9. Next, load the customization variables from a data set by selecting **L - Load variables from data set**.

10. Enter `NDCELL.NDNODEA.SAVECFG` as the data set name as shown in Figure 7-9 and press Enter to retrieve the stored variables. This data set is designated in the initial Portal configuration to store customization values.



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

Load Customization Variables

 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in:
    'BBWP6050.SEJPEXEC(EJPWVARS)'


 Data set name: 'NDCELL.NDNODEA.SAVECFG'


 If this data set is not cataloged, specify the volume.

 Volume:
```

Figure 7-9 Entering data set name to load customization variables

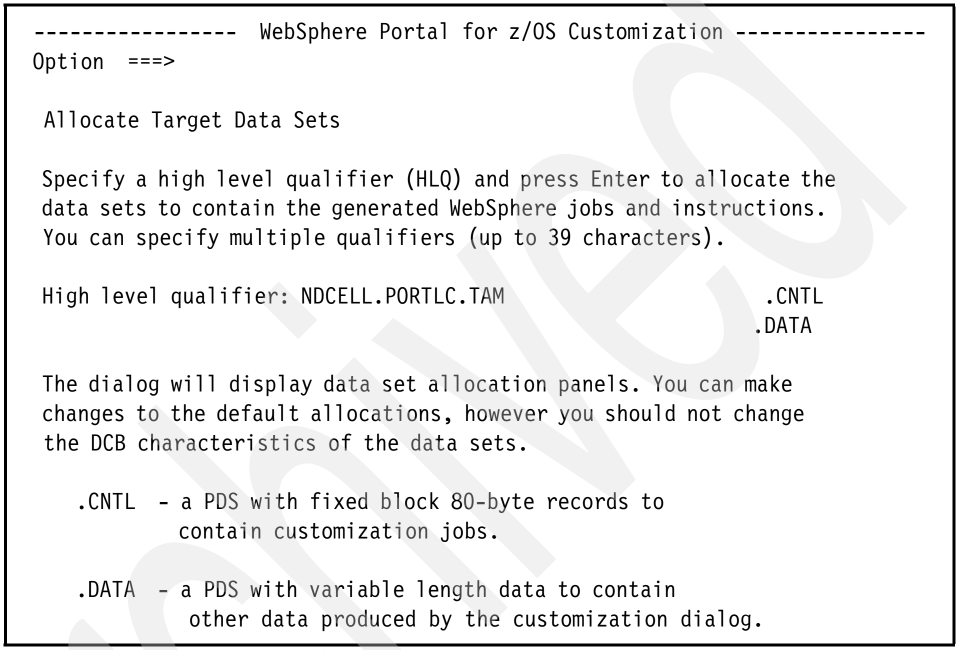
When you customize Portal, the variables that get defined and the generated jobs for that configuration task get stored to target data sets.

11. Select option **1 - Allocate target data sets**, the panel shown in Figure 7-10 on page 253 will appear.

Best practice for naming these target data sets is to choose a name that is reflective of the configuration task. In our example, we provide a data set name of NDCELL.PORTALC.TAM to represent that we are configuring Portal with TAM for our external security manager.

These data sets will store generated customization jobs (.CNTL) and other data provided by the customization panels (.DATA).

12. Provide a high level qualifier for the data set as shown in Figure 7-10.



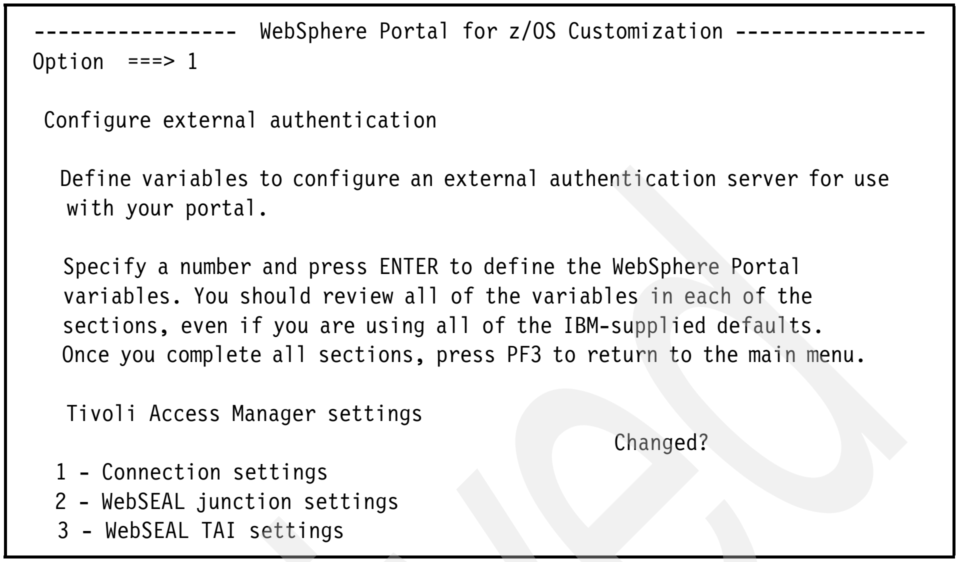
```
-----------------  WebSphere Portal for z/OS Customization ----------------
Option  ===>

 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: NDCELL.PORTLC.TAM                      .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 customization jobs.

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

Figure 7-10 Allocating target data sets for external authentication

13. Press Enter twice to accept the default .CNTL and .DATA data set parameters.

The previous steps are common to customizing any tasks with WebSphere Portal on z/OS. Now we access panels that are specific to the task at hand, which is configuring external authentication.

1. Select option **2 -Define variables for this specific task** and press Enter.
2. Figure 7-11 on page 254 is the initial panel for configuring authentication. Select option **1 - Connection settings** and press Enter.



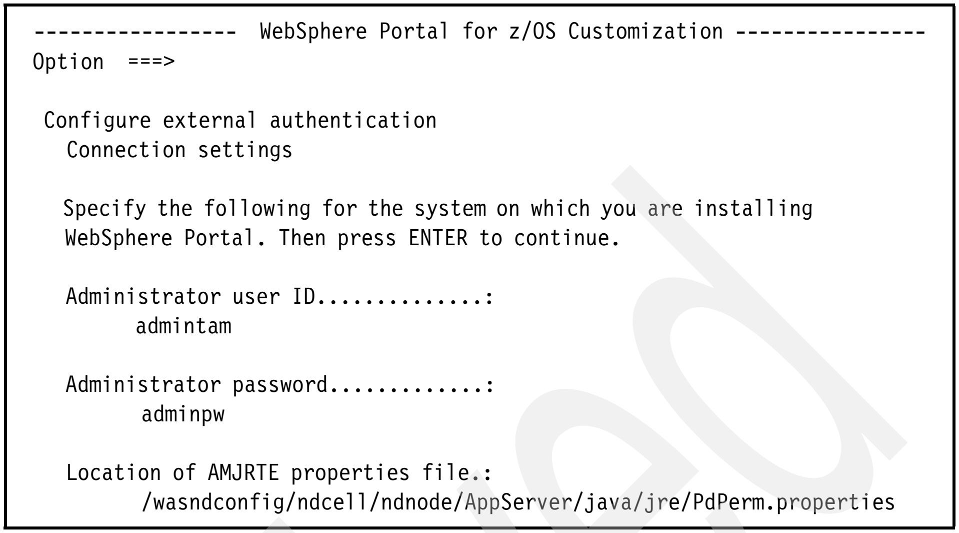
```
-----------------  WebSphere Portal for z/OS Customization ----------------
Option  ===> 1

 Configure external authentication

   Define variables to configure an external authentication server for use
    with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.

    Tivoli Access Manager settings
                                                     Changed?
  1 - Connection settings
  2 - WebSEAL junction settings
  3 - WebSEAL TAI settings
```

Figure 7-11 Selecting connection settings

3. Enter the TAM connection information as shown in Figure 7-12 on page 255. The connection settings are used to set up communication between WebSphere Portal on z/OS and TAM.
4. Table 7-1 provides a description of the input variables for connection settings.

Table 7-1 Input variables for Connection settings

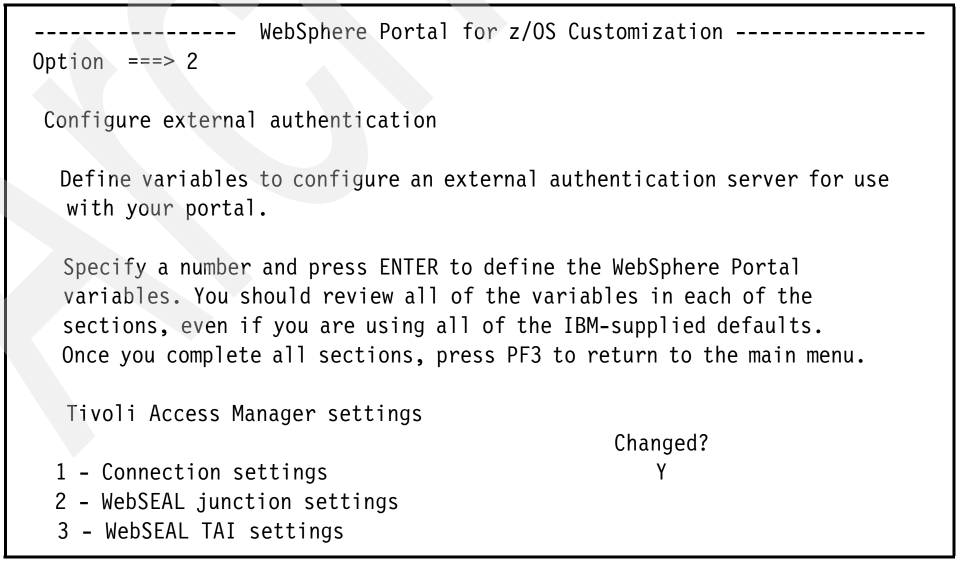
Input variables	Description
Administrator user ID and password	TAM administrator. Best practice is to not use the default TAM admin user, sec_master, but instead to create an admin ID that has the same authority as sec_master.
Location of AMJRTE properties file	The PdPerm.properties file gets created in this location by the Tivoli Access Manager SvrSslCfg command and contains information, such as: Policy Server host name, ports, version of AMJRTE, and path to encryption keys.



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

 Configure external authentication
   Connection settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Administrator user ID.............:
         admintam

   Administrator password............:
         adminpw

   Location of AMJRTE properties file.:
         /wasndconfig/ndcell/ndnode/AppServer/java/jre/PdPerm.properties
```

Figure 7-12 Providing connecting settings variables

5. Press Enter to return to the initial Configure external authentication panel. Also notice that `Y` is shown in the `Changed?` column to indicate that connection settings have been entered.
6. Select option **2 - WebSEAL junction settings**, as shown in Figure 7-13 and press Enter.



```
-----------------  WebSphere Portal for z/OS Customization ----------------
Option  ===> 2

 Configure external authentication

  Define variables to configure an external authentication server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.

   Tivoli Access Manager settings
                                                     Changed?
  1 - Connection settings                               Y
  2 - WebSEAL junction settings
  3 - WebSEAL TAI settings
```

Figure 7-13 Selecting WebSEAL junction settings

7. Enter the WebSEAL junction settings as shown in Figure 7-14 on page 257. The WebSEAL connection settings are used to create a connection in WebSEAL that will be recognized by WebSphere Application Server.

8. Table 7-2 lists descriptions of the input variables for WebSEAL junction settings.

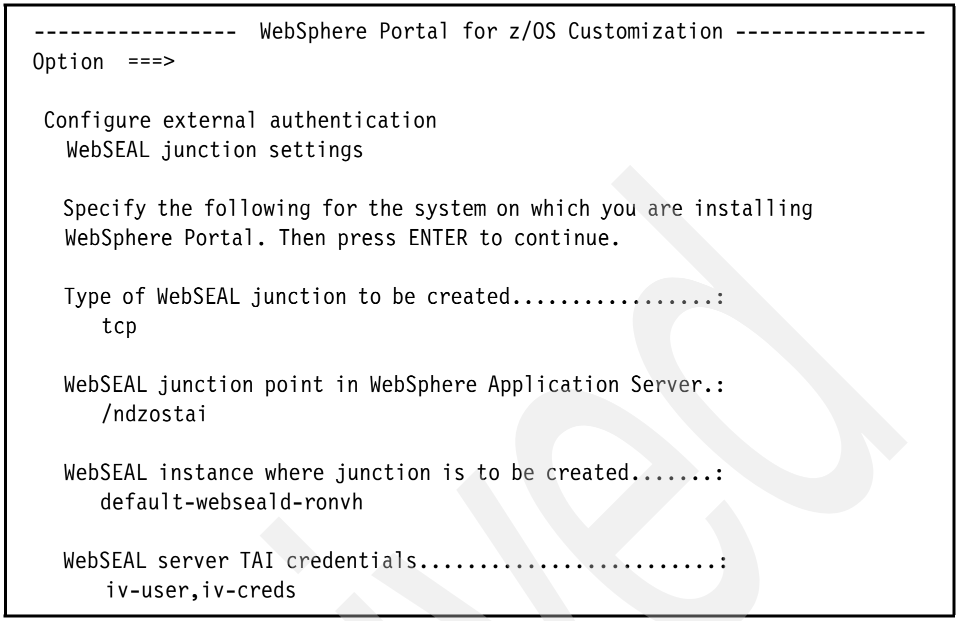
Table 7-2 Input variables for WebSEAL junction settings

Input variables	Description
Type of WebSEAL junction	Used to determine how WebSEAL will handle requests. There are two supported options: TCP and SSL.
WebSEAL junction point in WebSphere Application Server	Portal mount point in WebSEAL object space. Make sure the forward slash (/) is in front of the junction point.
WebSEAL instance	Specifies the WebSEAL server that will create the junction.
WebSEAL server TAI credentials	Used by TAI to identify the request originated from WebSEAL. If using TAM as an external authorization manager, you must include at least the iv-user and iv-creds headers.

9. Example 7-2 shows the command to use in Pdadmin to list WebSEAL instances.

Example 7-2 Server list command in pdadmin

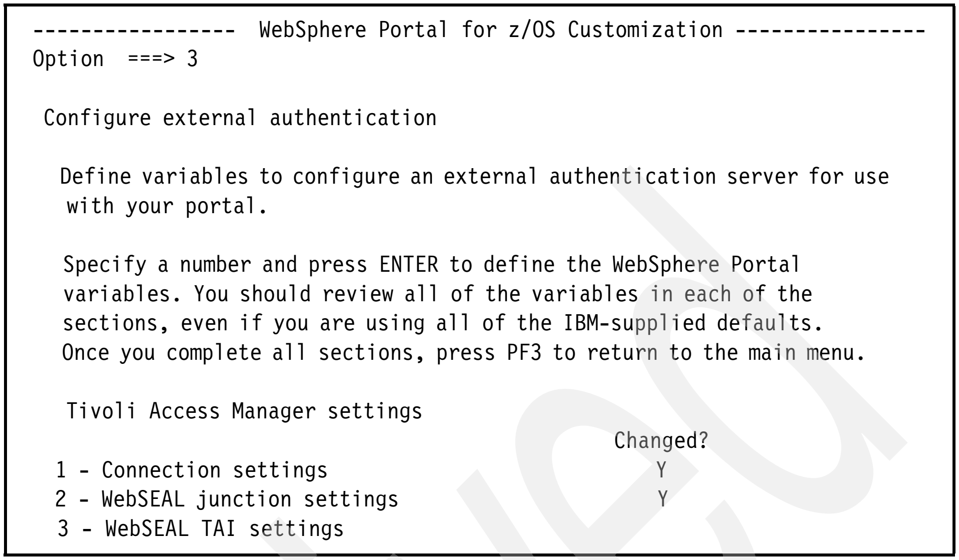
```
pdadmin> server list
default-webseald-ronvh
```



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

 Configure external authentication
   WebSEAL junction settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Type of WebSEAL junction to be created................:
      tcp

   WebSEAL junction point in WebSphere Application Server.:
      /ndzostai

   WebSEAL instance where junction is to be created.......:
      default-webseald-ronvh

   WebSEAL server TAI credentials.........................:
      iv-user,iv-creds
```

Figure 7-14 Providing WebSEAL junction settings variables

10. Press Enter to return to the initial Configure external authentication panel. Also notice that `Y` is shown in the `Changed?` column to indicate that WebSEAL junction settings have been entered.
11. Select option **3 - WebSEAL TAI settings**, as shown in Figure 7-15 on page 258.



```
-----------------  WebSphere Portal for z/OS Customization ----------------
Option  ===> 3

 Configure external authentication

   Define variables to configure an external authentication server for use
    with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.

    Tivoli Access Manager settings
                                                          Changed?
  1 - Connection settings                                     Y
  2 - WebSEAL junction settings                               Y
  3 - WebSEAL TAI settings
```

Figure 7-15   *Selecting WebSEAL TAI settings*

12. Enter the WebSEAL TAI settings as shown in Figure 7-16 on page 260. The WebSEAL TAI settings are used to enable authentication between WebSEAL and WebSphere Application Server.

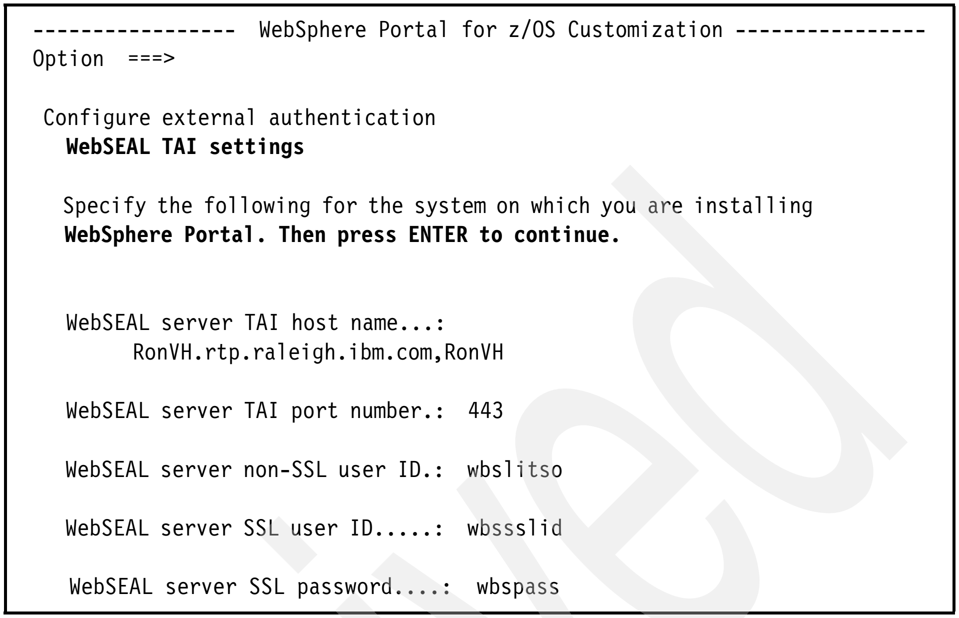
13. Table 7-3 lists descriptions of the input variables for WebSEAL TAI settings.

Table 7-3   Input variables for WebSEAL TAI settings

Input variables	**Description**
WebSEAL server TAI host name	TAI will handle requests from the listed servers. This value is case-sensitive. If you have more than one host name, then use commas to delimit the list. Must include both the fully qualified host name (RonVH.rtp.raleigh.ibm.com) and the network short name (RonVH).
WebSEAL port number	The port that WebSEAL server TAI listens on for requests. If using multiple ports, they must be comma-delimited.

Input variables	Description
WebSEAL server non-SSL user ID	The WebSEAL identity used in a TCP junction. WebSphere Application Server will use this identity to establish the “trust” that is required to validate the WebSEAL iv-* headers. The password for this ID should be set in the WebSEAL instance’s webseald.config on the basicauth-dummy-passwd property.
WebSEAL server SSL user ID	The WebSEAL identity used in a SSL junction. WebSphere Application Server will use this identity to establish the “trust” that is required to validate the WebSEAL iv-* headers.
WebSEAL server SSL password	The password associated with the SSL user ID.

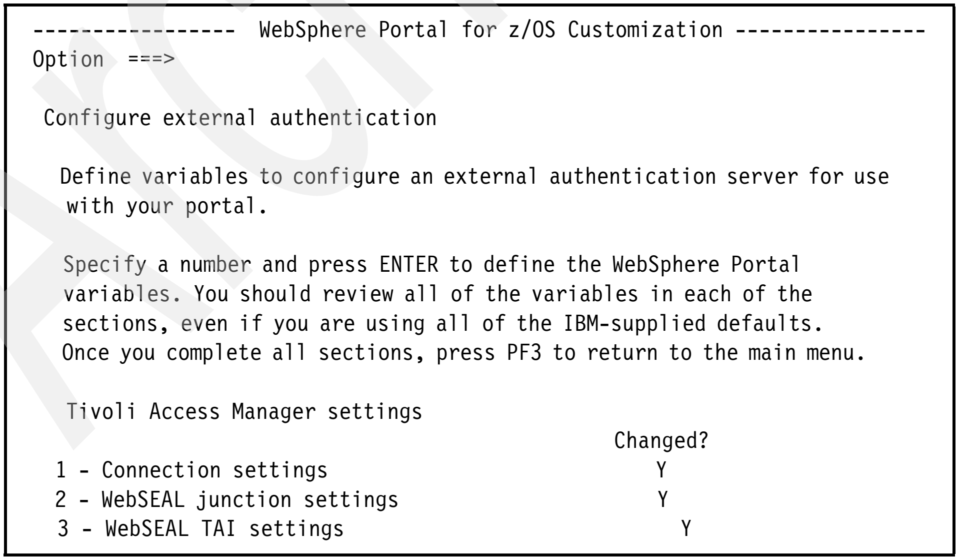
- **Notes:** The WebSEAL server non-SSL user ID and password should also be manually added as a user in Portal’s LDAP directory.
- The WebSEAL server SSL user ID and the WebSEAL server SSL password are required even when using a TCP junction.



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

 Configure external authentication
   WebSEAL TAI settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.


   WebSEAL server TAI host name...:
         RonVH.rtp.raleigh.ibm.com,RonVH

   WebSEAL server TAI port number.:  443

   WebSEAL server non-SSL user ID.:  wbslitso

   WebSEAL server SSL user ID.....:  wbssslid

   WebSEAL server SSL password....:  wbspass
```

Figure 7-16   Providing WebSEAL TAI settings variables

14. Press Enter to return to the initial Configure external authentication panel. Also notice that `Y` is shown in the `Changed?` column to indicate that WebSEAL TAI settings have been entered.



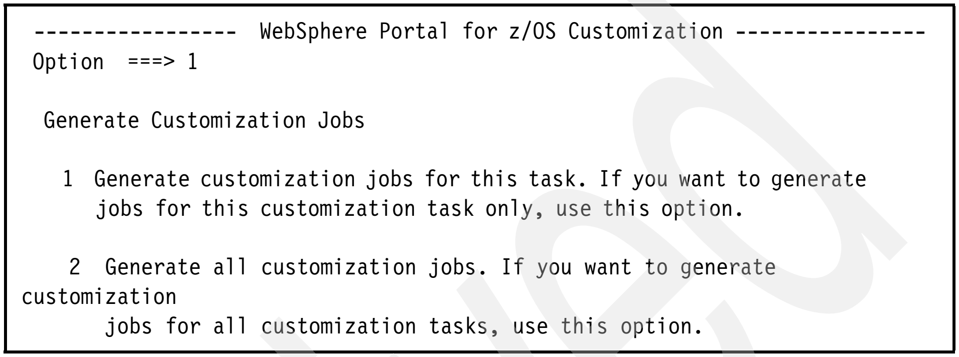
```
----------------  WebSphere Portal for z/OS Customization ----------------
Option  ===>

 Configure external authentication

   Define variables to configure an external authentication server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.

   Tivoli Access Manager settings
                                                   Changed?
  1 - Connection settings                             Y
  2 - WebSEAL junction settings                       Y
  3 - WebSEAL TAI settings                            Y
```

Figure 7-17   All Configure external authentication settings have been entered

15. Press F3 to exit the Define Variables dialog.

16. Select option **3 - Generate customization jobs** and press Enter.

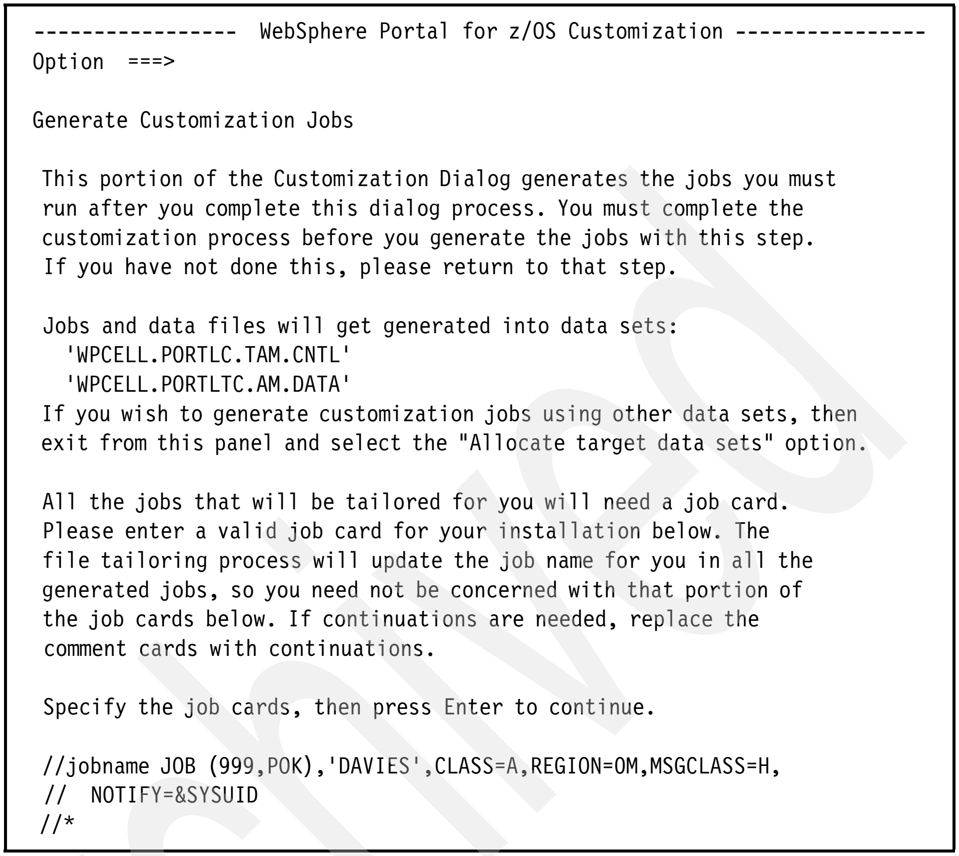
17. Select option **1 - Generate customization jobs for this task**, as shown in Figure 7-18 and press Enter. This is a new feature in IBM WebSphere Portal Enable for z/OS Version 6.0.0.1.



```
 -----------------  WebSphere Portal for z/OS Customization ----------------
 Option  ===> 1

  Generate Customization Jobs

    1  Generate customization jobs for this task. If you want to generate
       jobs for this customization task only, use this option.

     2  Generate all customization jobs. If you want to generate
customization
        jobs for all customization tasks, use this option.
```

Figure 7-18 Selecting Generate customization jobs for this task

18. Specify the job cards, shown in Figure 7-19 on page 262, and press Enter.



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

Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'WPCELL.PORTLC.TAM.CNTL'
   'WPCELL.PORTLTC.AM.DATA'
 If you wish to generate customization jobs using other data sets, then
 exit from this panel and select the "Allocate target data sets" option.

 All the jobs that will be tailored for you will need a job card.
 Please enter a valid job card for your installation below. The
 file tailoring process will update the job name for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations.

 Specify the job cards, then press Enter to continue.

 //jobname JOB (999,POK),'DAVIES',CLASS=A,REGION=0M,MSGCLASS=H,
 //  NOTIFY=&SYSUID
 //*
```

Figure 7-19   Generate Customized Jobs panel for configuring external authentication

Table 7-4 lists the generated .CNTL jobs that will be executed to configure external authentication, and also the purpose of the job.

Table 7-4   Description of generated .CNTL jobs

Generated job	Description
EJPIESA	Contains the generated instructions.
EJPSESO1	Used to configure an SSL connection between TAM and portal.
EJPSESOV	Used to verify connectivity to TAM.
EJPSESA	Used to configure WebSEAL Authentication server for use with portal.

Generated job	Description
EJPSESAP	(Optional) Used to enable automatic user provisioning to TAM. After this feature is enabled, users created in Portal are automatically imported into TAM.

19. Press Enter to return to the panel Generate Customization Jobs.
20. Select **S - Save customization variables** and press Enter.

Specify the name of the sequential data set that contains the environment configuration variables. If the data set does not exist, the dialog displays the Allocate New Data Set panel.

In our case, we store our environment configuration variables in a data set named NDCELL.NDNODEA.SAVECFG.

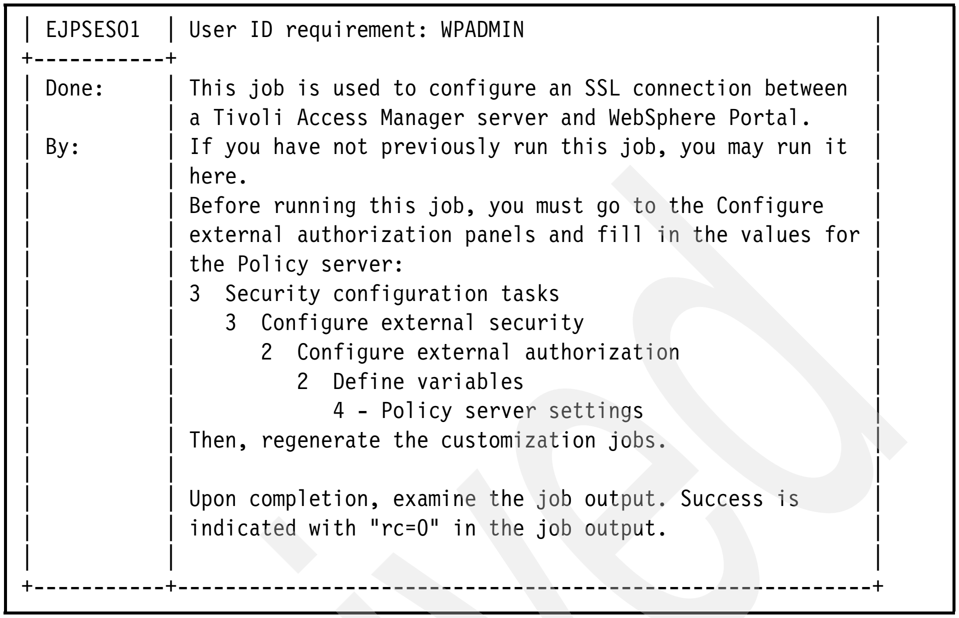
1. Enter `NDCELL.NDNODEA.SAVECFG` as the data set name and press Enter.

Viewing the generated instructions will tell us which jobs to run.

2. Before executing the generated jobs, select option **4 - View instructions** and press Enter.

Return to the Portal customization dialog to configure the communication between the TAM Policy Server and Portal.

3. Before running job EJPSESO1, the values for the Policy Server must be entered as noted in Figure 7-20 on page 264.



```
| EJPSESO1  | User ID requirement: WPADMIN                             |
+-----------+                                                          |
| Done:     | This job is used to configure an SSL connection between  |
|           | a Tivoli Access Manager server and WebSphere Portal.     |
| By:       | If you have not previously run this job, you may run it  |
|           | here.                                                    |
|           | Before running this job, you must go to the Configure    |
|           | external authorization panels and fill in the values for |
|           | the Policy server:                                       |
|           | 3  Security configuration tasks                          |
|           |    3  Configure external security                        |
|           |       2  Configure external authorization                |
|           |          2  Define variables                             |
|           |             4 - Policy server settings                   |
|           | Then, regenerate the customization jobs.                 |
|           |                                                          |
|           | Upon completion, examine the job output. Success is      |
|           | indicated with "rc=0" in the job output.                 |
|           |                                                          |
+-----------+----------------------------------------------------------+
```

Figure 7-20 Instructions to follow before running EJPSESO1

4. Press F3 to return to exit the generated instructions.
5. Press F3 to return to the initial Configure external authentication panel.

Policy Server connection settings is a task associated with Configure external authorization. Proceed as follows:

1. Select option **2 - Configure external authorization** and press Enter.
2. Select option **2 - Define variables** and press Enter.
3. Select option **4 - Policy server settings** and press Enter.
4. Enter the Policy Server settings as shown in Figure 7-21 on page 265.

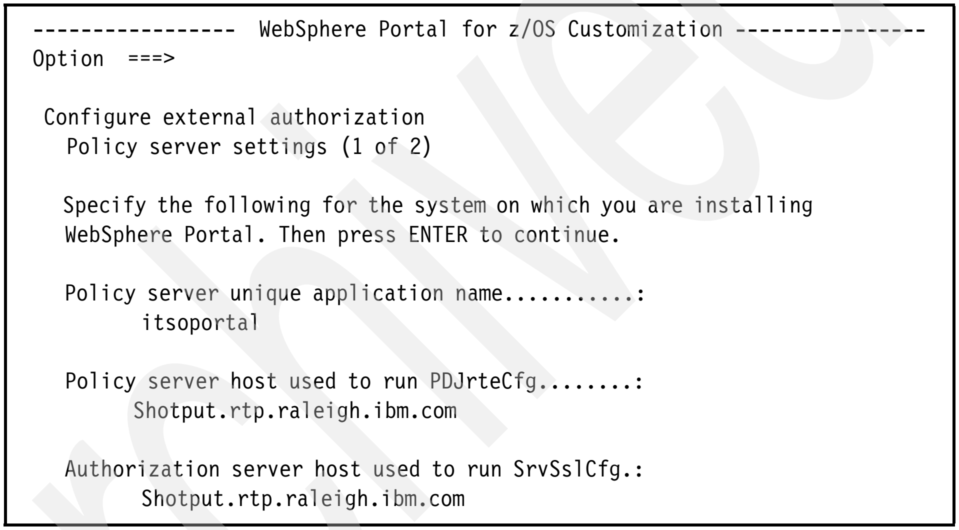
   Table 7-5 lists descriptions of the variables for Policy Server settings.

Table 7-5 Input variables for Policy Server settings

Input variables	Description
Policy server unique application name	Used to create a new server in Policy server.

Input variables	Description
Policy server host	The fully qualified name of the Policy server that is used to run PDJrteCfg, which is the TAM Java runtime component. This value can also be an IP address.
Authorization server host	The fully qualified name of the Authorization Server that is used to run SrvSslCfg.

Figure 7-12 on page 255 illustrates entering the Policy server settings variables.



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

 Configure external authorization
   Policy server settings (1 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Policy server unique application name...........:
          itsoportal

   Policy server host used to run PDJrteCfg........:
         Shotput.rtp.raleigh.ibm.com

   Authorization server host used to run SrvSslCfg.:
          Shotput.rtp.raleigh.ibm.com
```

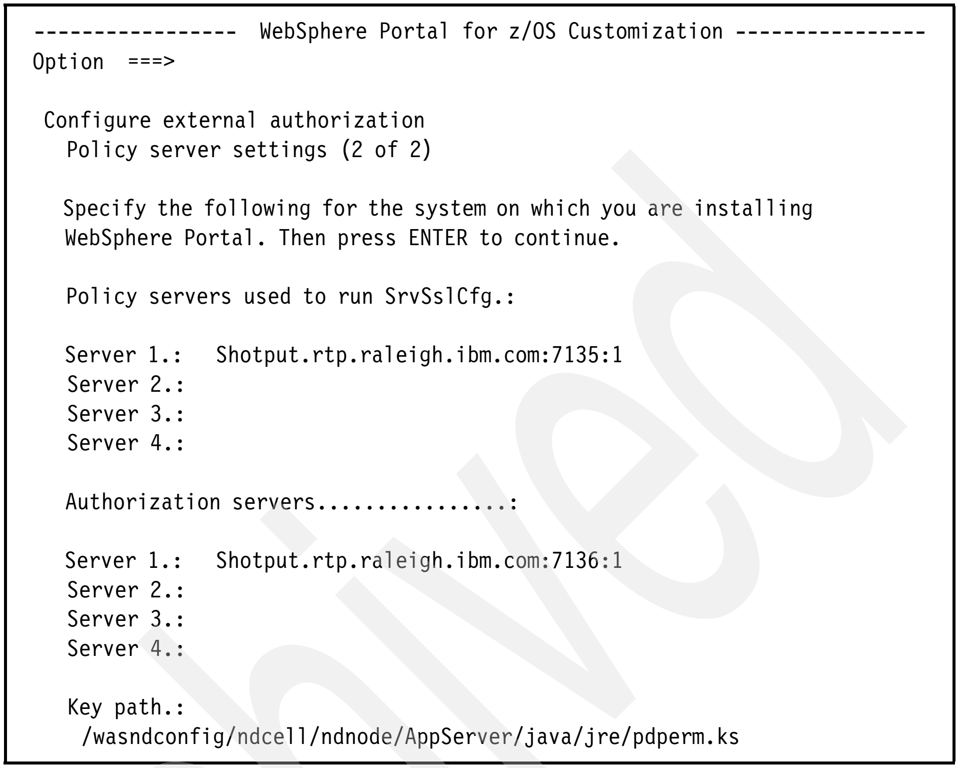
Figure 7-21   Providing Policy server settings variables

5. Press Enter to enter the Policy server variables on the second panel as shown in Figure 7-22 on page 267.

   Table 7-6 on page 266 lists descriptions of input variables for Policy server settings.

Table 7-6 Input variables for second panel of Policy server settings

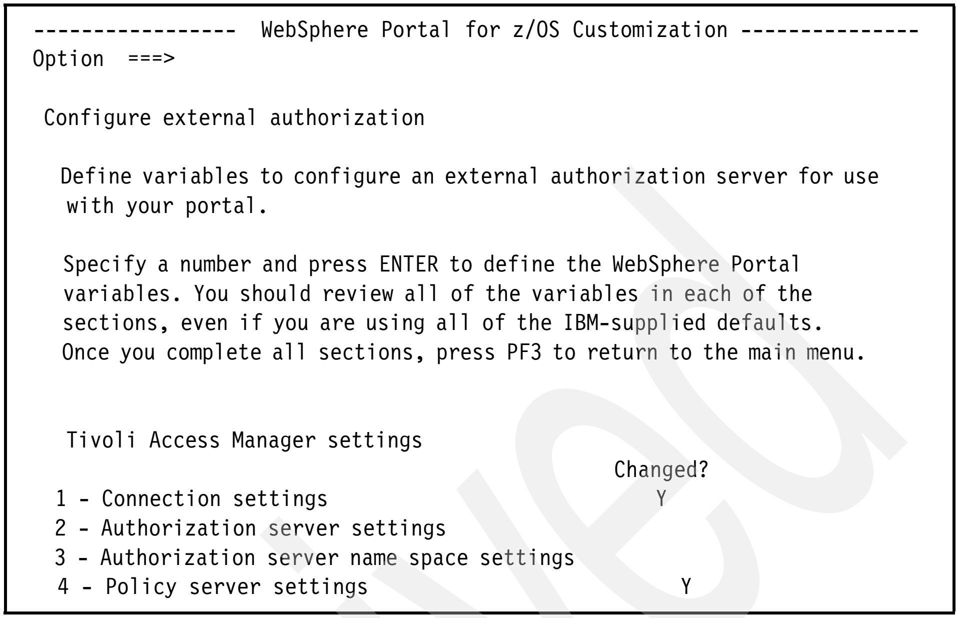
Input variables	Description
Policy servers used to run SrvSslCfg	This name is a combination of the fully qualified host name or IP address of the Policy server, the port that the Policy uses to listen for requests, and the priority of the Policy server. List these servers by priority.
Authorization server	This name is a combination of the fully qualified host name or IP address of the Authorization server, the port that the Authorization uses to listen for requests, and the priority of the Authorization Policy. List these servers by priority.
Key path	Location for storing the encryption keys, which are used for the SSL communication between AMJRTE and TAM. This file is generated as a result of the SrvSslCfg command. This is the absolute path of the Policy Director key.



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

 Configure external authorization
   Policy server settings (2 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Policy servers used to run SrvSslCfg.:

   Server 1.:   Shotput.rtp.raleigh.ibm.com:7135:1
   Server 2.:
   Server 3.:
   Server 4.:

   Authorization servers...............:

   Server 1.:   Shotput.rtp.raleigh.ibm.com:7136:1
   Server 2.:
   Server 3.:
   Server 4.:

   Key path.:
    /wasndconfig/ndcell/ndnode/AppServer/java/jre/pdperm.ks
```

Figure 7-22 Providing Policy server settings variables

6. Press Enter to return to the initial Configure external authorization panel. Also notice that `Y` is shown in the `Changed?` column to indicate that Policy server settings have been entered, as shown in Figure 7-23 on page 268.



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

 Configure external authorization

   Define variables to configure an external authorization server for use
    with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


    Tivoli Access Manager settings
                                                 Changed?
  1 - Connection settings                            Y
  2 - Authorization server settings
  3 - Authorization server name space settings
  4 - Policy server settings                          Y
```

Figure 7-23 All Policy server settings have been entered

The next steps are to generate the customization jobs for this task (option **3**). Save the customization variables (option **S**). View the instructions to understand in which order to run the generation jobs (option **4**).

Note: Be sure to run one job at a time. Make sure rc=0 is received for each job before proceeding to run the next job.

Verifying external authentication

In this section we list the steps for verifying that WebSphere Portal on z/OS has been successfully configured for external authentication with TAM.

Use the TAM Web Portal Manager (TAM WPM) to verify that the WebSEAL junction was created.

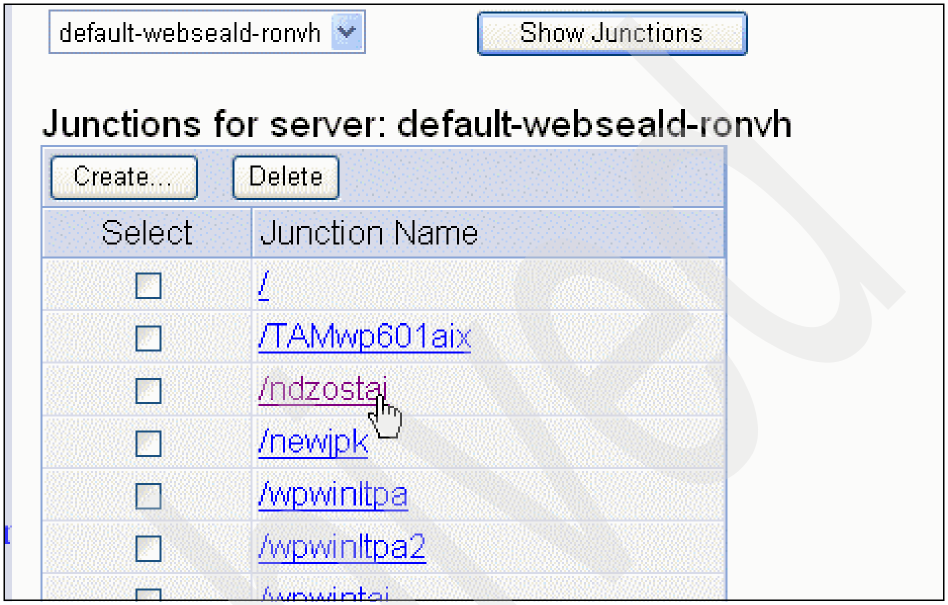
1. In TAM WPM, on the Task List, click **WebSEAL -> List Junctions**.

   Junctions are managed by WebSEAL servers to which they have been configured. Our newly created junction is managed by default-webseald-ronvh WebSEAL server.

2. In Manage Junctions, select your WebSEAL server name from the drop-down list.

In our case, we designated /ndzostai as the junction point in the WebSEAL junction settings panel.

3. In Manage Junctions, click **Show** Junctions as shown in Figure 7-24. Verify that the `/ndzostai` is in the list.



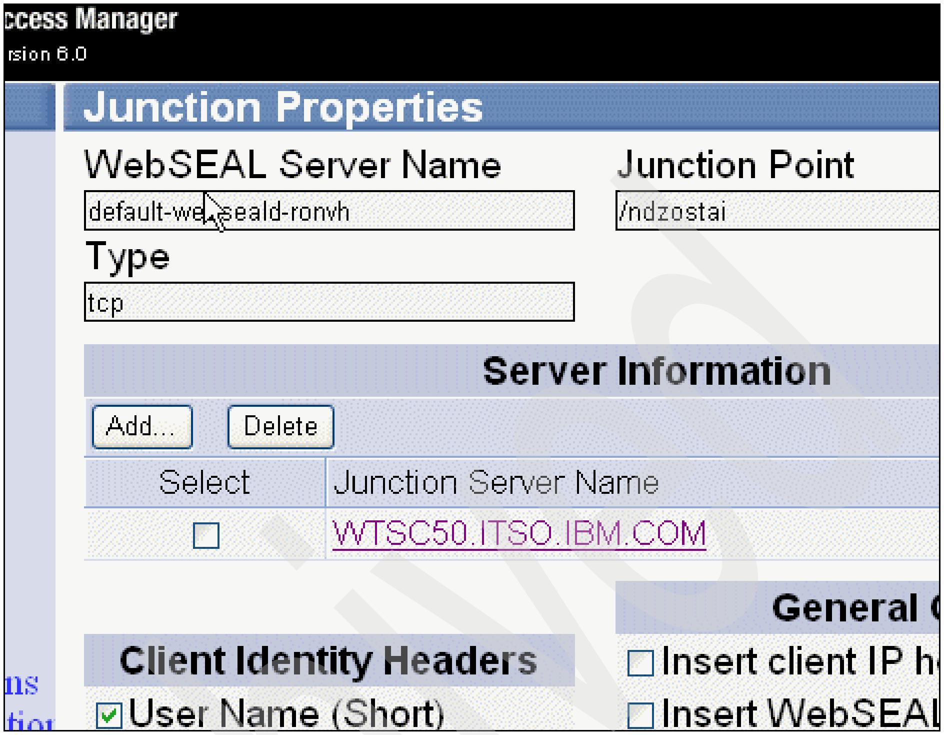
Junctions for server: default-webseald-ronvh

Select	Junction Name
□	/
□	/TAMwp601aix
□	/ndzostai
□	/newjpk
□	/wpwinltpa
□	/wpwinltpa2
□	/wpwintai

Figure 7-24 WebSEAL junction list

4. Select **/ndzostai** to view the junction properties.

5. Notice that the Junction Server Name is the fully qualified Portal host name and the Type is `tcp`, as shown in Figure 7-25 on page 270.



Junction Properties

WebSEAL Server Name: default-webseald-ronvh

Junction Point: /ndzostai

Type: tcp

Server Information

Select	Junction Server Name
□	WTSC50.ITSO.IBM.COM

General (

□ Insert client IP h

□ Insert WebSEAL

Client Identity Headers

☑ User Name (Short)

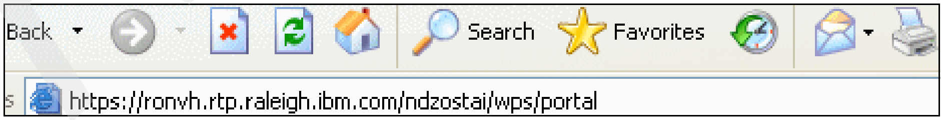
Figure 7-25 Junction properties

6. Sign off from TAM WPM by clicking **Sign Off** in the lower right corner of page.

After verifying that the configuration was set up properly in TAM, test that you can access the Portal with the WebSEAL junction point and bypass the Portal login page, and that you are presented with the Portal personalized home page.

7. Open a Web browser and enter the following in the address field, as shown in Figure 7-26:

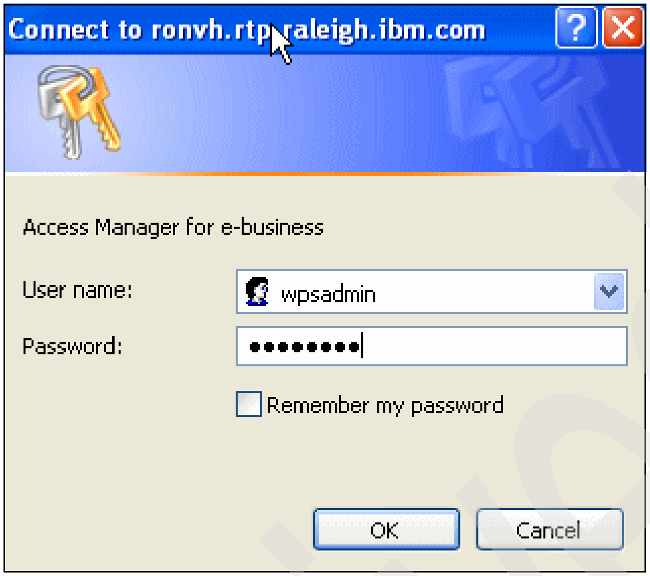
   ```
   https://WebSEAL_hostname:WebSEAL_port/junction/wps/<Policy_server_un
   ique_application_name>
   ```



https://ronvh.rtp.raleigh.ibm.com/ndzostai/wps/portal

Figure 7-26 Accessing Portal with new WebSEAL junction

8. Authenticate WebSEAL challenge with a user that is in TAM WebSEAL's LDAP, as shown in Figure 7-27 on page 271.



Connect to ronvh.rtp.raleigh.ibm.com

Access Manager for e-business

User name: wpsadmin

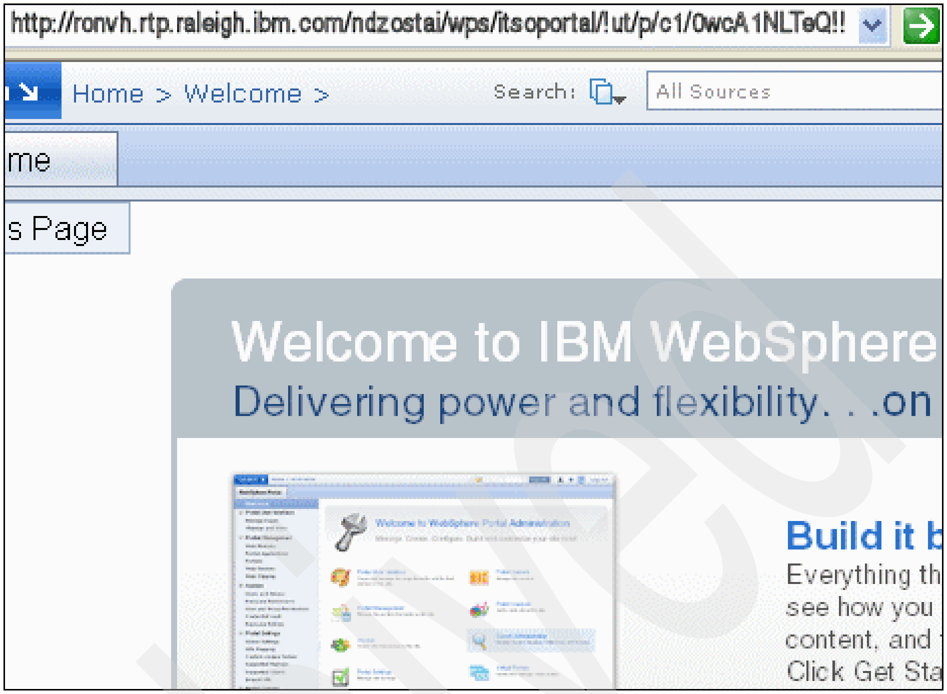
Password: ●●●●●●●●●

□ Remember my password

Figure 7-27 WebSEAL authentication challenge

9. After you authenticate WebSEAL, you will bypass the Portal login screen and be directed to the secure and personalized Portal page, as shown in Figure 7-28 on page 272.

   In this case, our <portal_personalized_context_root> is itsoportal.



http://ronvh.rtp.raleigh.ibm.com/ndzostai/wps/itsoportal/!ut/p/c1/0wcA1NLTeQ!!

Welcome to IBM WebSphere

Delivering power and flexibility. . .on

Build it b

Everything th
see how you
content, and
Click Get Sta

Figure 7-28 Secured and personalized Portal page

Errors encountered

This section describes the problems encountered and our solutions when configuring WebSphere Portal on z/OS with an external security manager.

Problem

EJPSESO1 build failed due to a SOAP connection error. When we did not run the generated jobs under the WebSphere Application Server Admin user, we saw errors as shown in Figure 7-29 on page 273 in the output of job EJPSESO1.

Note, however, that the jobs are generated from the dialogs with the WebSphere Application Server admin user ID and pw into the job card.

```
action-init-cfg-files-zos:
Tue Apr 10 20:18:19 EDT 2007

action-update-portal-properties:
Ýportalproperty¨ Tue Apr 10 20:18:19 EDT 2007
  Ýwsadmin¨ WASX7023E: Error creating "SOAP" connection to host
"wtsc50.itso.ibm
agement.exception.ConnectorNotAvailableException: ÝSOAPException:
faultCode=SOAP
ketException; targetException=java.lang.IllegalArgumentException:
Error opening
  Ýwsadmin¨ WASX7213I: This scripting client is not connected to a
server proces
/ndnode/AppServer/profiles/default/logs/wsadmin.traceout for
additional informat
  Ýwsadmin¨ WASX8011W: AdminTask object is not available.
```

Figure 7-29 Soap connection error

Solution

Run the generated jobs under the WebSphere Application Server Admin user where indicated to resolve the SOAP connection error. This is because when WebSphere Global security is enabled, communication with the Deployment Manager over SOAP will take place over SSL. In order for the SSL handshake to complete, the “client” (which in this case is the user ID that the batch job is running under) must possess a keyring that contains the CA certificate being used by the WebSphere cell.

Passing -userid and -password properties is not sufficient.

Problem

EJPSESA build failed due to a communication error. Figure 7-30 displays the errors we saw in the output of job EJPSESA.

```
BUILD FAILED
file:../config/actions/esm_cfg.xml:558: Exception occurred. Ensure
the default-webseald-ronvh WebSeal instance is running:
Ý
HPDBA0207E   A communication error occurred while initializing the
SSL connection.
```

Figure 7-30 Job EJPSESA output

Solution

Ensure that WebSEAL is running and WebSphere Portal on z/OS can access it. In our configuration, we had to pass through firewalls. Make sure the inbound and outbound ports are open on the Portal on the z/OS server and WebSEAL servers.

Problem

Accessing the Portal with WebSEAL junction gets you back to Portal log in page instead of bypassing this page and accessing the personalized home page.

Solution

Ensure that the WebSEAL server non-SSL and SSL users are added to the user registry that is integrated with the Portal. The WebSEAL server non-SSL and SSL user IDs are used to represent the WebSEAL identity to WebSphere Application Server.

7.3.2 External authorization using Tivoli Access Manager

When security is enabled and does not use TAM, access control to WebSphere Portal resources are administered internally by WebSphere Portal. Configuring external authorization allows you to use an external security manager to control access to some or all WebSphere Portal resources.

As documented in the InfoCenter, WebSphere Portal authorization is role-based. When you enable external authorization, WebSphere Portal can externalize roles and uses ACLs to control role membership. After a role has been externalized, the external security manager only sees the externalized role as containing one permission: membership in the role. WebSphere Portal still determines the permissions associated with the role. For more detail, see the topic External Authorization in the InfoCenter.

Resources can be moved back and forth from internal to external control using the Resource Permissions portlet in WebSphere Portal. Explicit role assignments are preserved when moving in either direction. However, inherited role memberships are blocked for externalized resources. When you externalize access control for a resource, that resource can only be administered through the external security manager interface.

After externalization, role membership can only be assigned and removed using the external security manager. The only action that can be taken by the Resources Permissions portlet is to move the object back to internal control.

There are limitations on externalizing WebSphere Portal resources (refer to the InfoCenter for more detail):

- Private pages cannot be externalized.
- If you externalize a resource using the Resource Permissions portlet, access control for all public child resources move with it. Using the XML configuration interface (xmlaccess) does not have the same behavior.
- After you externalize access control for a resource, you must use the external security manager to assign users to roles for the resource.
- You can use either the Resources Permissions portlet or the XML configuration interface to create additional role types on a resource after the access control is externalized.
- Externalizing the access control for a resource severs any access control inheritance from any internally-controlled parent resources.

Configuring external authorization

In our test environment, we used Tivoli WebSEAL Proxy and Tivoli Access Manager. In this step we configure WebSphere Portal authentication to use external authorization. We already have WebSEAL and TAM configured for external authentication (see 7.3.1, “External authentication using Tivoli Access Manager and the Trust Association Interceptor” on page 249).

External authentication is required before configuring external authorization because we need to be able to verify a user’s identify before we can determine if they have permission to access Portal resources.

Note: You must have configured external authentication before configuring external authorization.

At this point we were building on the previous configuration steps. The basic portal configuration was complete, the DB2 transfer had been run, security had been configured and enabled, and external authentication had been configured and enabled. We were now ready to configure external authorization. We used some of the definitions from previous steps and some new definitions unique to this configuration step.

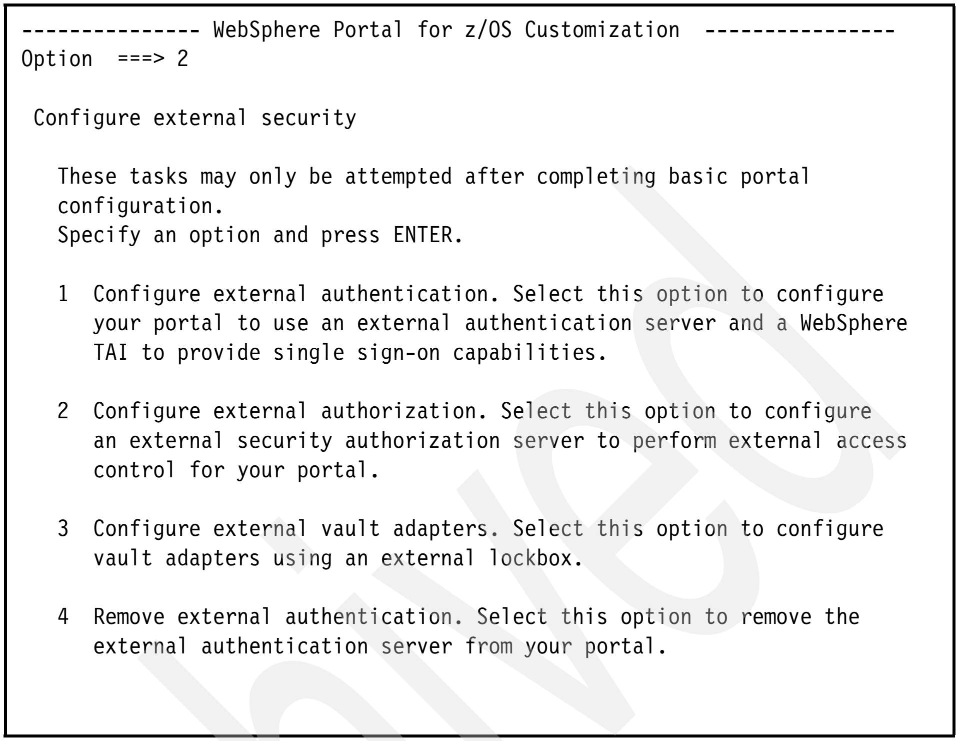
Note: You must have completed the basic configuration tasks before configuring security.

After we configured external authentication using TAM, we used the following steps to configure external authorization using TAM. (Note that these steps repeat some of the steps we performed previously in other configuration tasks.)

1. Logon to TSO and select ISPF/Program Development Facility from the Master Application Menu.
2. Select option **6 - Command** from the ISPF Primary Option Menu.
3. Enter the following command to start the WebSphere Portal configuration dialog

   ```
   exec 'BBWP6049.SBBOCLIB(BBOWSTRT)' 'APPL(PS1) PROD(EJP)
   PRODHLQ(BBWP6049)
   ```

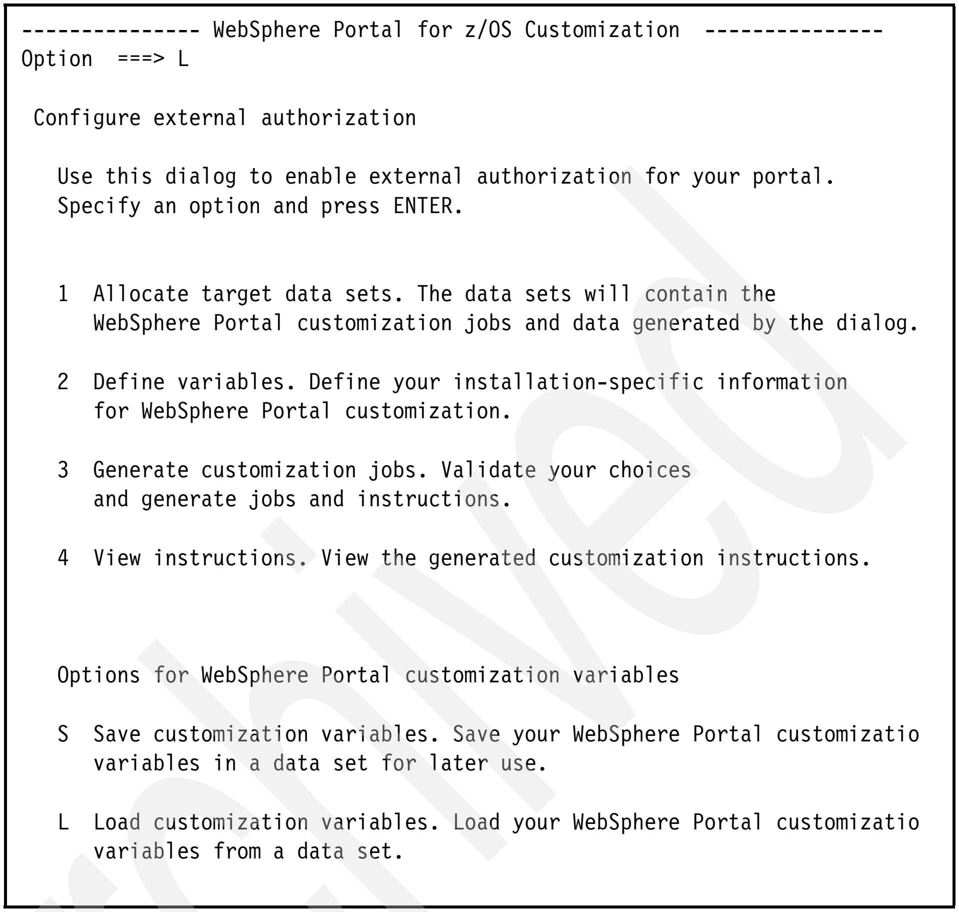
4. The first panel shown is WebSphere Application Server for z/OS Customization. Press Enter to continue.
5. Select option **5 - WebSphere Application Server-based add-on products** on the next WebSphere Application Server for z/OS Customization panel. We are configuring WebSphere Portal which is an add-on product to WebSphere Application Server for z/OS.
6. Select option **1 - WebSphere Portal for z/OS Add-On Product Configuration**.
7. The next panel is the WebSphere Portal for z/OS Customization. Press Enter to continue.
8. Select option **3 - Security configuration tasks** on the Portal configuration panel.
9. Select option **3 - Configure external security** on the Security configuration tasks panel.
10. We had previously configured external authentication. Select option **2 - Configure external authorization**, as shown in Figure 7-31 on page 277.



```
--------------- WebSphere Portal for z/OS Customization  ----------------
Option  ===> 2

 Configure external security

   These tasks may only be attempted after completing basic portal
   configuration.
   Specify an option and press ENTER.

   1  Configure external authentication. Select this option to configure
      your portal to use an external authentication server and a WebSphere
      TAI to provide single sign-on capabilities.

   2  Configure external authorization. Select this option to configure
      an external security authorization server to perform external access
      control for your portal.

   3  Configure external vault adapters. Select this option to configure
      vault adapters using an external lockbox.

   4  Remove external authentication. Select this option to remove the
      external authentication server from your portal.
```

Figure 7-31 Configure external security - Configure external authorization

11. Load the customization variables previously saved. Select option **L** from the Options for WebSphere Portal customization variables panel, as shown inFigure 7-32 on page 278.

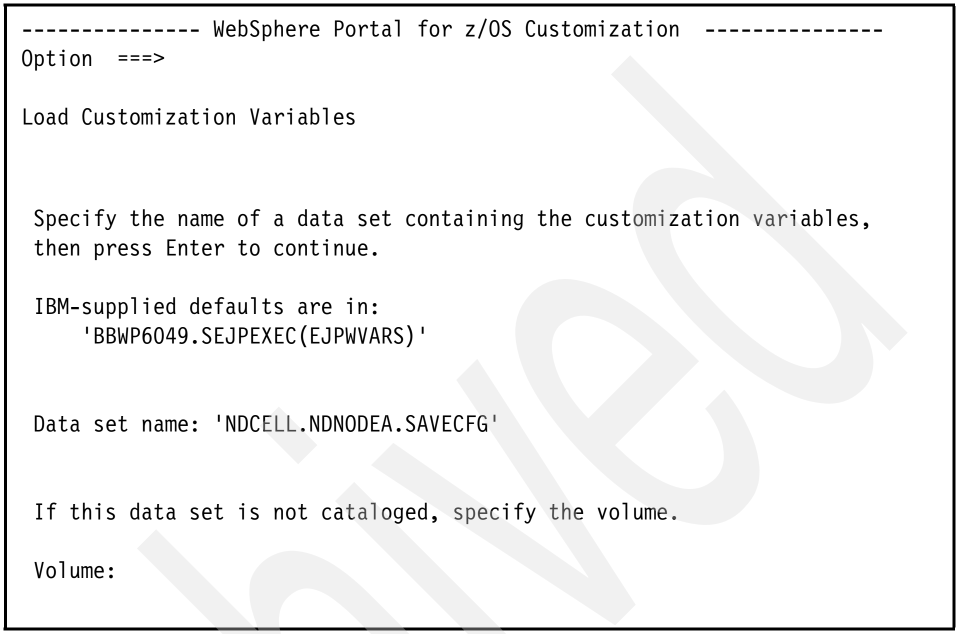


```
--------------- WebSphere Portal for z/OS Customization  ---------------
Option  ===> L

 Configure external authorization

   Use this dialog to enable external authorization for your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.



   Options for WebSphere Portal customization variables

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

   L  Load customization variables. Load your WebSphere Portal customizatio
      variables from a data set.
```

Figure 7-32 Configure external authorization - Load customization variables

Load the customization variables before you start to configure external authorization.

12. Load the customization variables previously saved. Verify the data set name and enter the volume if this is not a cataloged data set. The panel is shown in Figure 7-33.



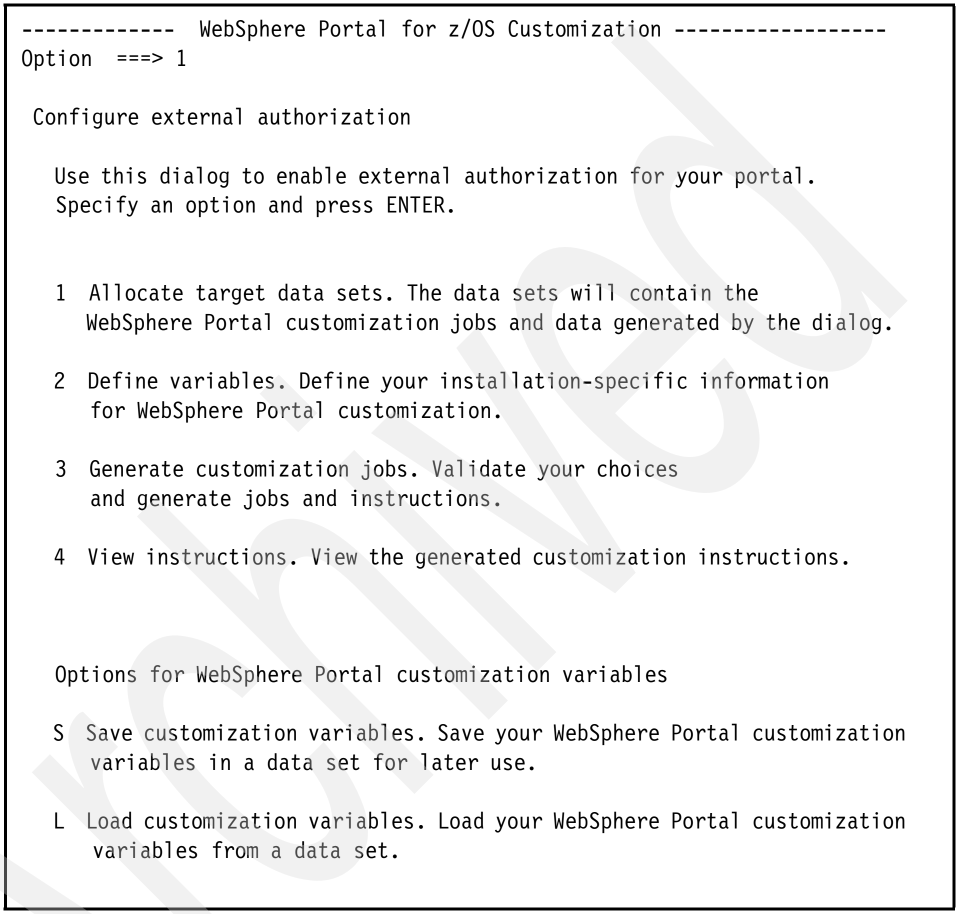
```
--------------- WebSphere Portal for z/OS Customization  ---------------
Option  ===>

Load Customization Variables



 Specify the name of a data set containing the customization variables,
 then press Enter to continue.

 IBM-supplied defaults are in:
     'BBWP6049.SEJPEXEC(EJPWVARS)'


 Data set name: 'NDCELL.NDNODEA.SAVECFG'


 If this data set is not cataloged, specify the volume.

 Volume:
```

Figure 7-33 Loading the Customization Variables

13. The panel displays `NDCELL.NDNODEA.SAVECFG` as the data set name. This data set is defined in the initial portal configuration to store customization values (see Figure 7-9 on page 252).

    Press Enter to retrieve the stored variables.

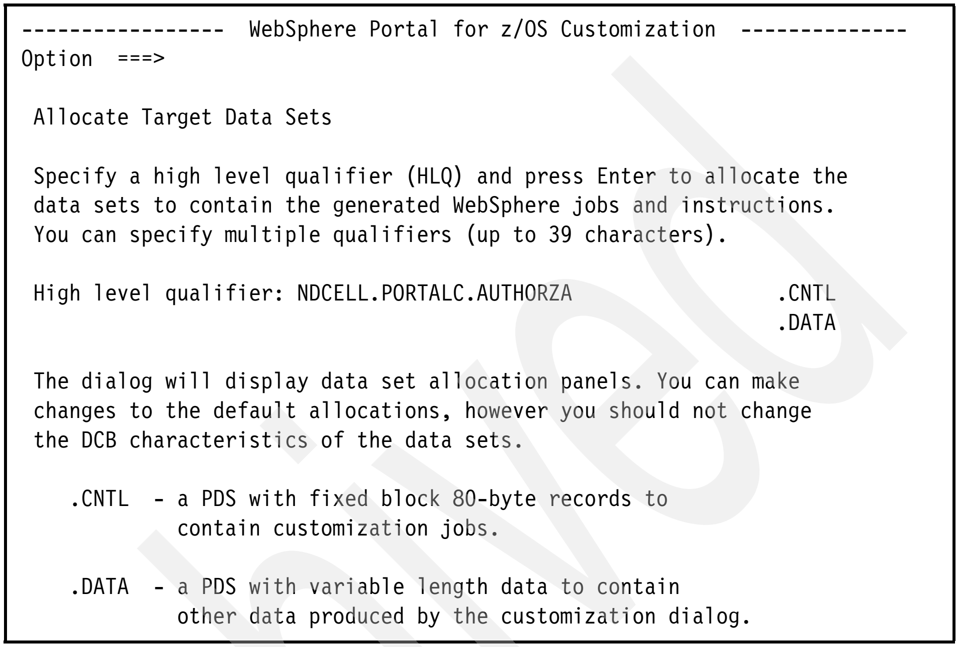
14. You need to allocate new target data sets where the JCL members and data members will be generated. This ensures that previous configuration JCL and data members are not overlaid. Select option **1 - Allocate target data sets**, as shown in Figure 7-34.



```
-------------  WebSphere Portal for z/OS Customization ------------------
Option  ===> 1

 Configure external authorization

   Use this dialog to enable external authorization for your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.



   Options for WebSphere Portal customization variables

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 7-34 Configure external authorization - Allocate target data sets

15.Enter the high level qualifier of the .CNTL and .DATA data sets. We used `NDCELL.PORTALC.AUTHORZA` to differentiate from previous configuration steps; see Figure 7-35.



```
-----------------  WebSphere Portal 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: NDCELL.PORTALC.AUTHORZA                 .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 customization jobs.

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

Figure 7-35 Configure external authorization - Allocate target data sets (continued)

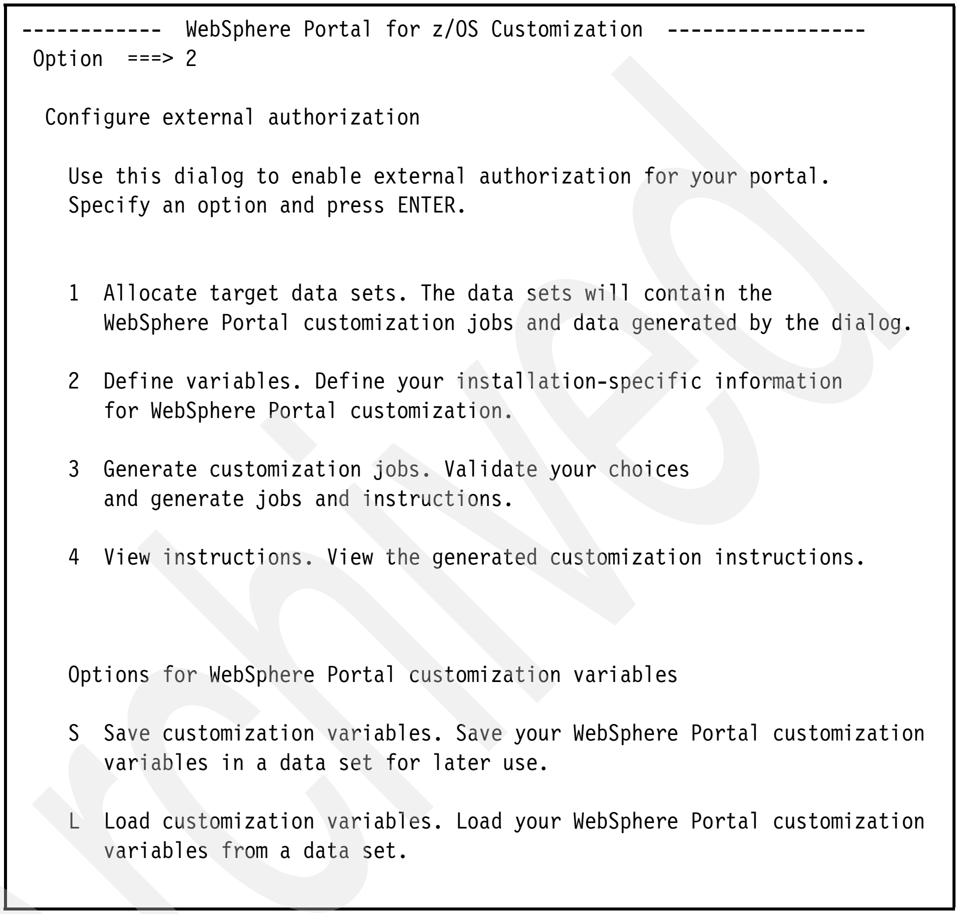
> **Note:** We recommend using a data set name that reflects the configuration task being performed. In this case we used NDCELL.PORTALC.AUTHORZA.

16.We used the default values to allocate both data sets. To do this, press Enter to continue to the second panel, and press Enter on the second panel.

After the data sets are allocated, the next steps are to define the variables required for the TAM external authorization. We used the planning worksheets from the InfoCenter and got some of the values we needed from the TAM administrator. See the InfoCenter for these worksheets, under topic “Configure external authorization” at:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/cu_cdsecextnauz_zos.html`

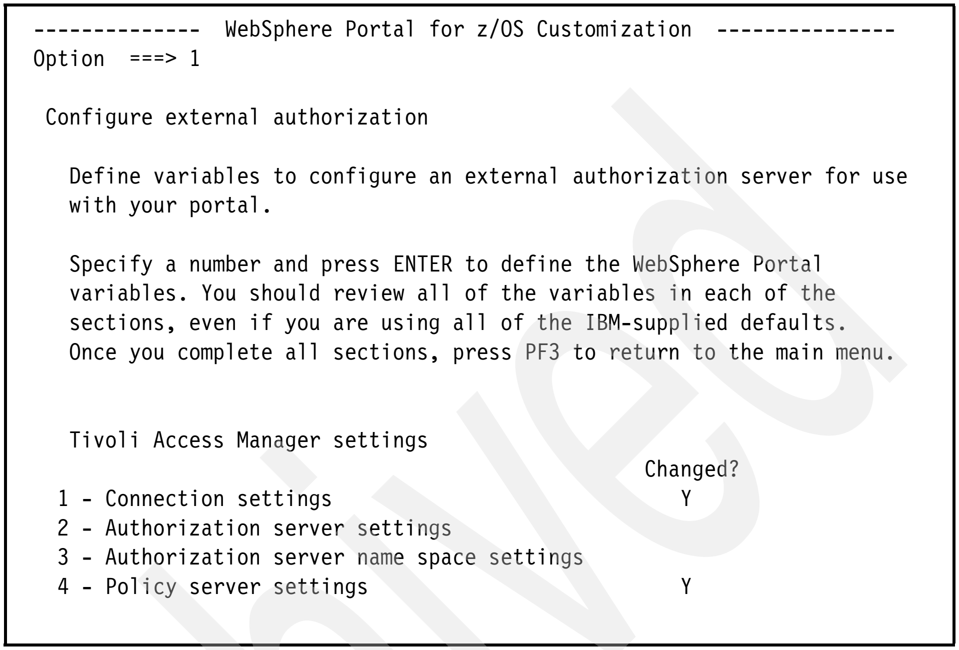
17. Define the variables for the external authorization configuration. These values are based on the TAM servers we used and were obtained from the TAM administrator. Select option **2 - Define variables**, as shown in Figure 7-36.



```
------------  WebSphere Portal for z/OS Customization  -----------------
 Option  ===> 2

  Configure external authorization

    Use this dialog to enable external authorization for your portal.
    Specify an option and press ENTER.


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

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

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

    4  View instructions. View the generated customization instructions.



    Options for WebSphere Portal customization variables

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

    L  Load customization variables. Load your WebSphere Portal customization
       variables from a data set.
```

Figure 7-36 Configure external authorization - Define variables

18. Begin to define the TAM settings. First define the TAM connection settings. Select option **1 - Connection settings**, as shown in Figure 7-37.

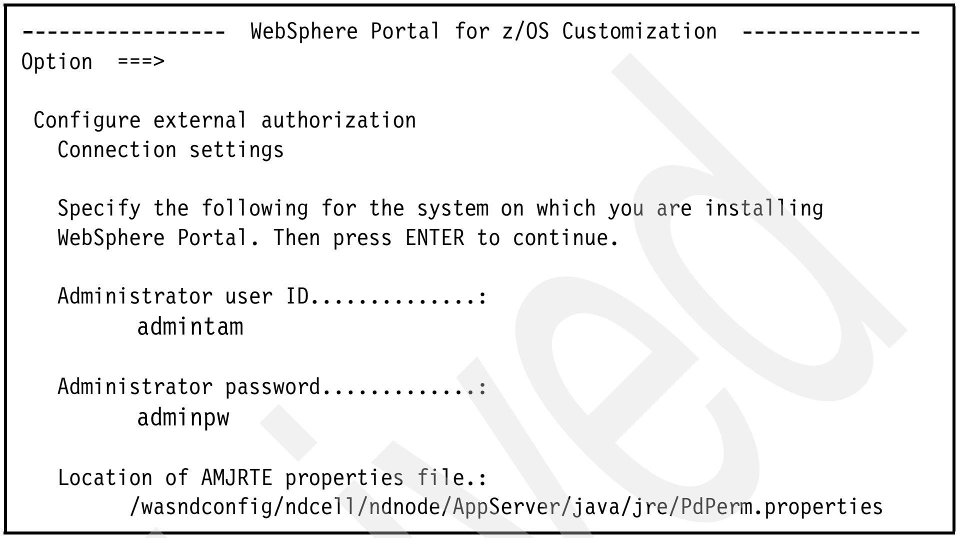


```
--------------  WebSphere Portal for z/OS Customization  ---------------
Option  ===> 1

 Configure external authorization

   Define variables to configure an external authorization server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


   Tivoli Access Manager settings
                                                     Changed?
  1 - Connection settings                              Y
  2 - Authorization server settings
  3 - Authorization server name space settings
  4 - Policy server settings                           Y
```

Figure 7-37 Configure external authorization - Connection settings

Note: Under the `Changed ?` column, some of the values will be `Y` if they were previously changed. They will be blank if they have not yet been changed.

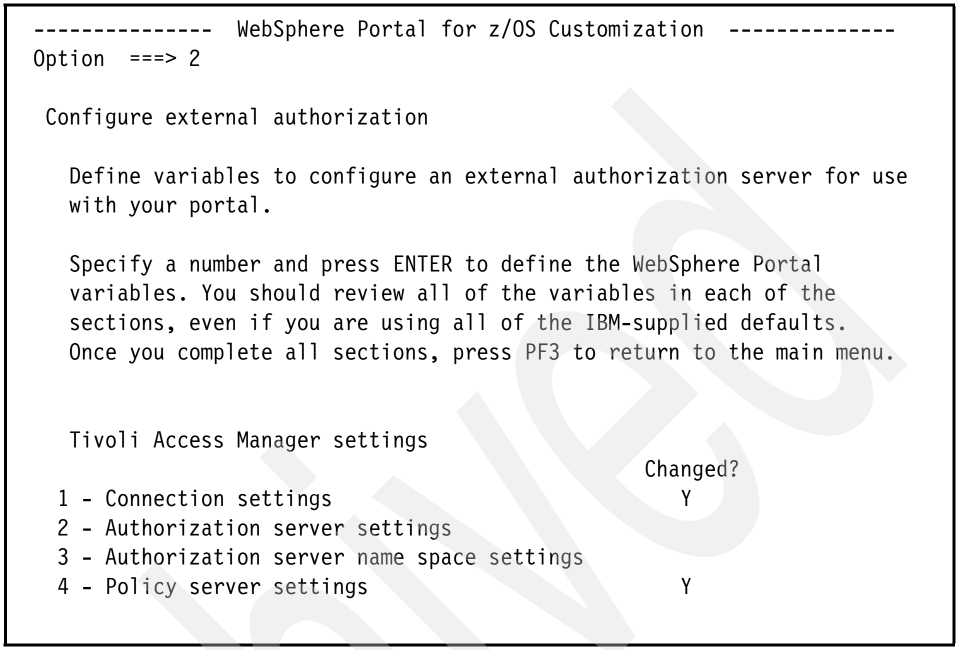
19. The connection settings values are the same values used to configure external authentication, as we were using the same TAM server. See Figure 7-12 on page 255 and Table 7-1 on page 254 for a description of the connection variables.



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

 Configure external authorization
   Connection settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Administrator user ID.............:
         admintam

   Administrator password............:
         adminpw

   Location of AMJRTE properties file.:
         /wasndconfig/ndcell/ndnode/AppServer/java/jre/PdPerm.properties
```

Figure 7-38   Configure external authorization - Connection settings (continued)

20. Now begin the new definitions. Select option **2 - Authorization server settings**, as shown in Figure 7-39.



```
---------------  WebSphere Portal for z/OS Customization  --------------
Option  ===> 2

 Configure external authorization

   Define variables to configure an external authorization server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


   Tivoli Access Manager settings
                                                  Changed?
  1 - Connection settings                            Y
  2 - Authorization server settings
  3 - Authorization server name space settings
  4 - Policy server settings                         Y
```

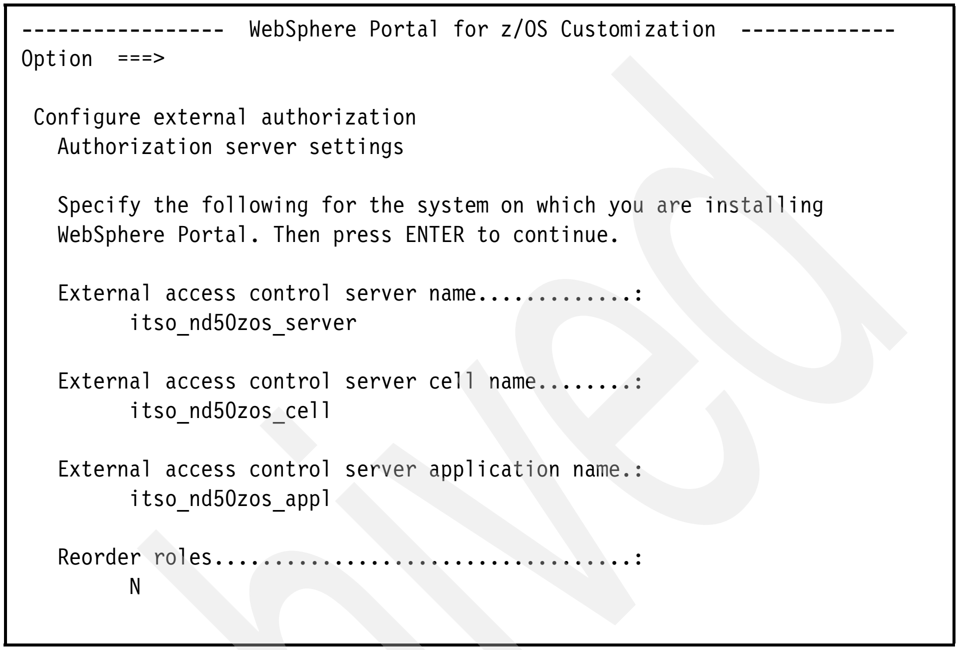
Figure 7-39 Configure external authorization - Authorization server settings

21. Table 7-7 lists the TAM connection settings from the Authorization Server settings worksheet filled out during planning.

Table 7-7 TAM connection settings worksheet

Input variable	**Value in the dialog after you load IBM defaults**	**Our values**
External access control server name	WebSphere_Portal	itso_nd50zos_server
External access control server cell name		itso_nd50zos_cell
External access control server application name	WPS	itso_nd50zos_appl
Reorder roles	N	N

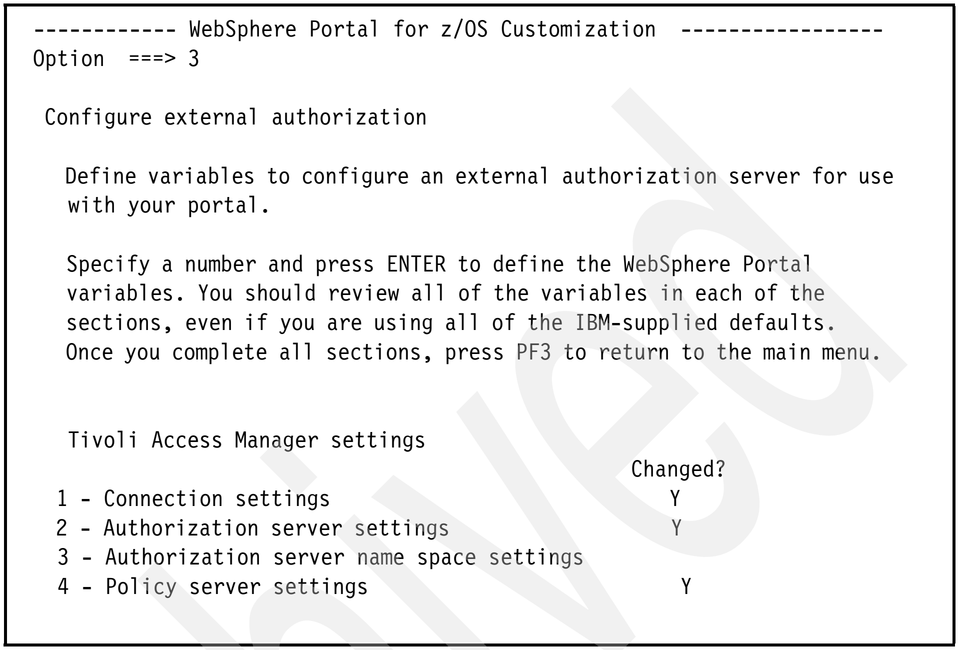
22. Configure the Authorization server settings. Enter the External access control server name, server cell name, and server application name. These will be created in TAM. See Figure 7-40.



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

 Configure external authorization
   Authorization server settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   External access control server name............:
         itso_nd50zos_server

   External access control server cell name........:
         itso_nd50zos_cell

   External access control server application name.:
         itso_nd50zos_appl

   Reorder roles...................................:
         N
```

Figure 7-40   Configure external authorization - Authorization server settings

23. Configure the authorization server namespace settings. Select option **3 - Authorization server name space settings**, as shown in Figure 7-41.



```
------------ WebSphere Portal for z/OS Customization  -----------------
Option  ===> 3

 Configure external authorization

   Define variables to configure an external authorization server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


   Tivoli Access Manager settings
                                                   Changed?
  1 - Connection settings                             Y
  2 - Authorization server settings                   Y
  3 - Authorization server name space settings
  4 - Policy server settings                          Y
```

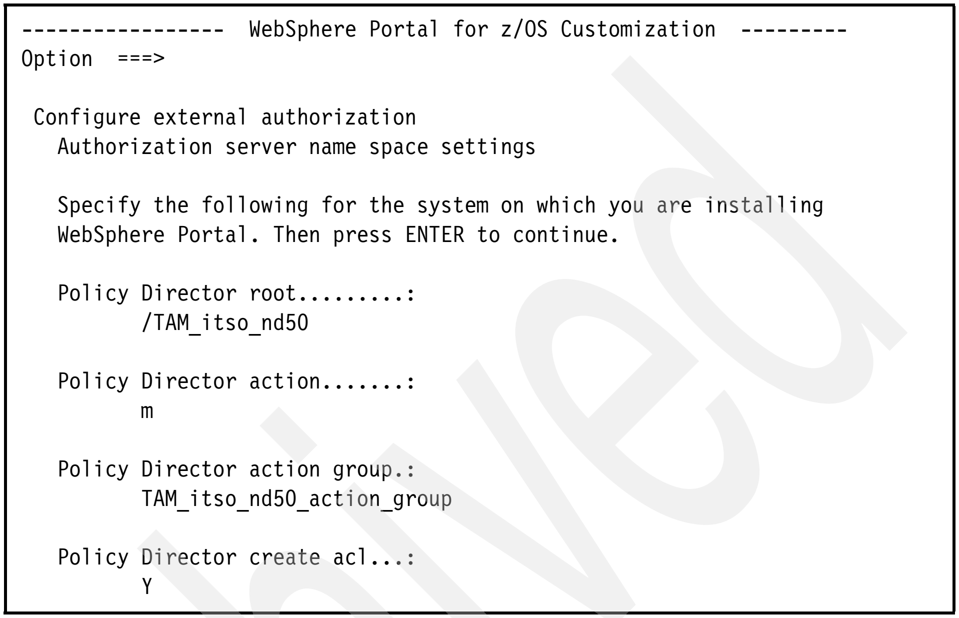
Figure 7-41 Configure external authorization - Authorization server name space settings

24. Table 7-8 shows the TAM authorization server namespace settings from the Authorization server namespace settings worksheet.

Table 7-8 Authorization server namespace settings worksheet

Item	Value in the Dialog after you load IBM defaults	Our value
Policy Director root	WPSv60	TAM_itso_nd50
Policy Director action	m	m
Policy Director action group	ÝWP6¨	TAM_itso_nd50_action_group
Policy Director create ACL	Y	Y

25. Define the authorization server namespace settings. Enter the values from the Authorization server namespace settings worksheet, as shown in Figure 7-42.

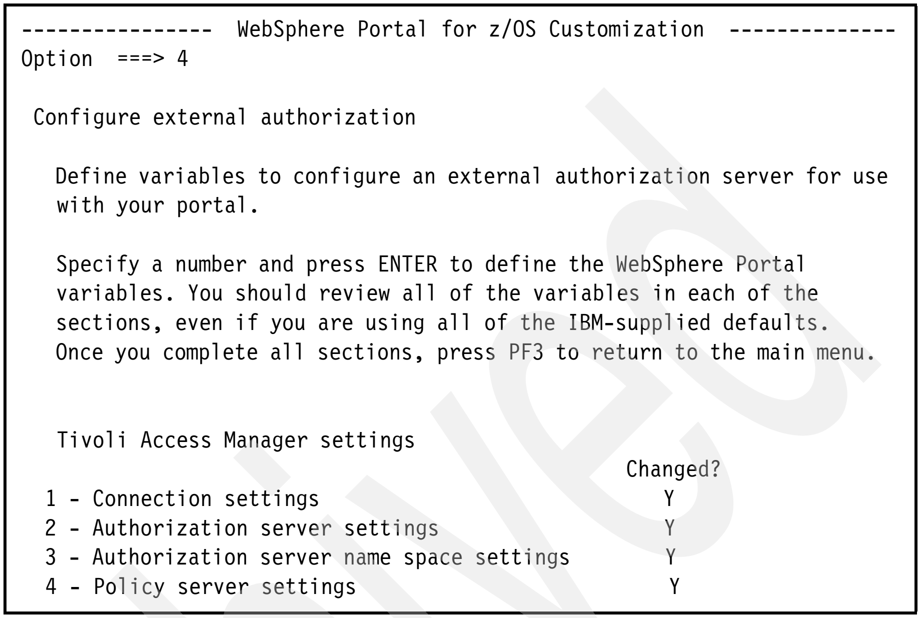


```
-----------------  WebSphere Portal for z/OS Customization  ---------
Option  ===>

 Configure external authorization
   Authorization server name space settings

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Policy Director root.........:
          /TAM_itso_nd50

   Policy Director action.......:
          m

   Policy Director action group.:
          TAM_itso_nd50_action_group

   Policy Director create acl...:
          Y
```

Figure 7-42 Configure external authorization - Authorization server name space settings

> **Note:** The default value for the Policy Director action group is enclosed in special characters on our screen. These characters may display differently based on your display code page settings. They are actually brackets and we found that if they were typed over or removed, the brackets are added back in when this definition is created in TAM.

26. Define the policy server settings. Select option **4 - Policy server settings**, as shown in Figure 7-43.



```
----------------  WebSphere Portal for z/OS Customization  --------------
Option  ===> 4

 Configure external authorization

   Define variables to configure an external authorization server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


   Tivoli Access Manager settings
                                                         Changed?
  1 - Connection settings                                   Y
  2 - Authorization server settings                         Y
  3 - Authorization server name space settings              Y
  4 - Policy server settings                                Y
```

Figure 7-43 Configure external authorization - Policy server settings

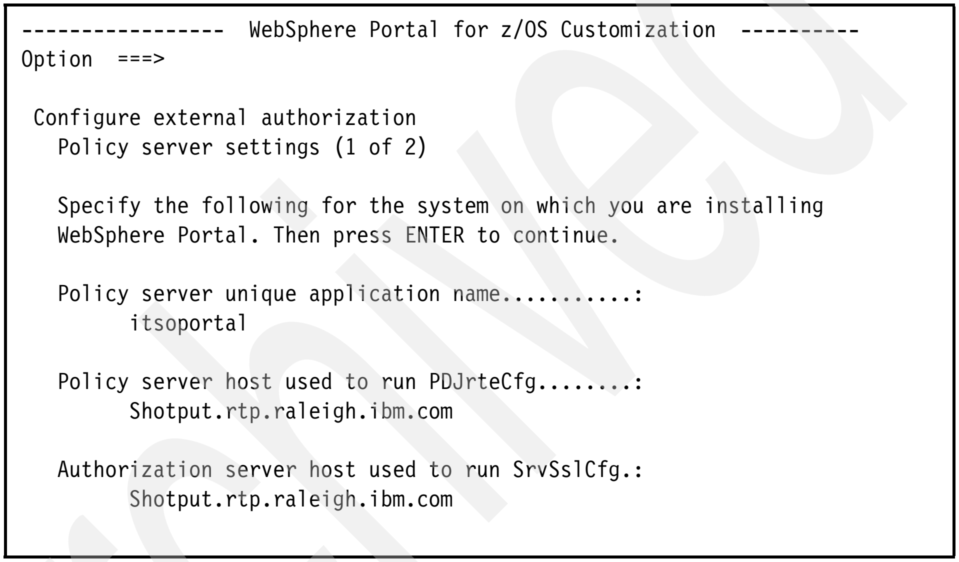
27. Table 7-9 lists the TAM Policy Server settings from the Policy server settings worksheet.

Table 7-9 Policy Server settings worksheet

Item	Values in the Dialog after you load IBM defaults	Our value
Policy server unique application name	amwp6	`itsoportal`
Policy server host used to run PDJrteCfg	your.TAM.Policy.Server.hostname	`Shotput.rtp.raleigh.ibm.com`
Authorization server host used to run SrvSslCfg	your.TAM.Authz.Server.hostname	`Shotput.rtp.raleigh.ibm.com`
Policy server used to run SrvSslCfg	your.TAM.Policy.server.hostname:7135:1	`Shotput.rtp.raleigh.ibm.com:7135:1`
Authorization servers	your.TAM.Policy.server.hostname:7136:1	`Shotput.rtp.raleigh.ibm.com:7136:1`

Item	Values in the Dialog after you load IBM defaults	Our value
Key path	/java/jre/lib/security/pdperms.ks	`/wasndconfig/ndcell/ndnode/AppServer/java/jre/pdperm.ks`

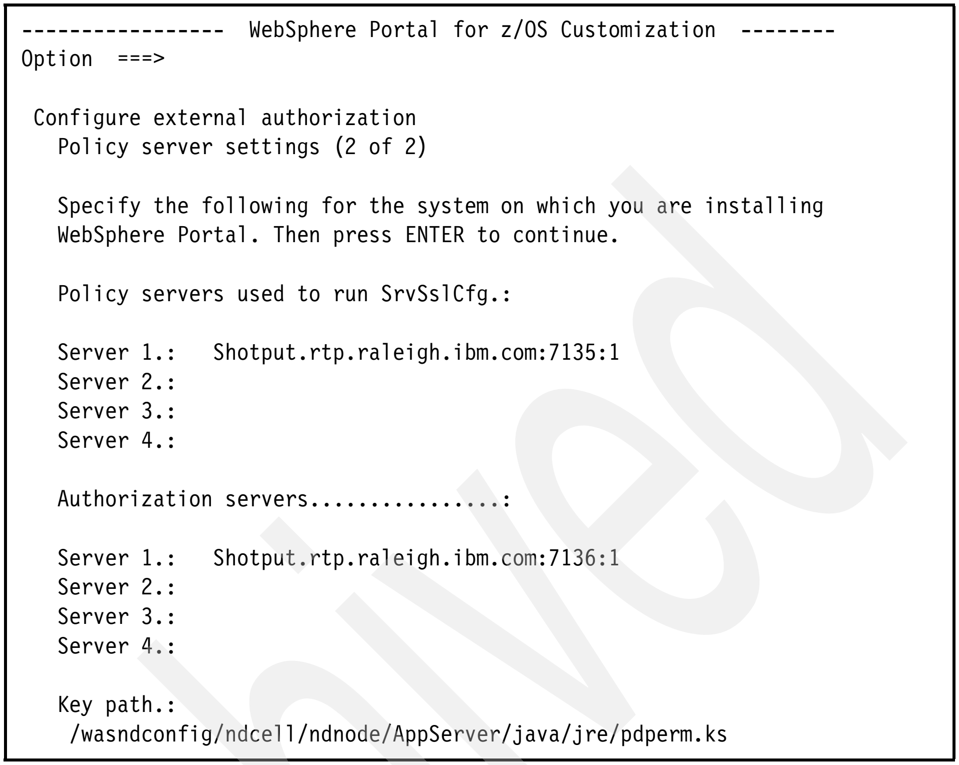
28. Enter the TAM Policy server settings. These values are from the Policy server settings worksheet. There are two panels for this step; Figure 7-44 displays the first part of the Policy server settings.



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

 Configure external authorization
   Policy server settings (1 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Policy server unique application name...........:
         itsoportal

   Policy server host used to run PDJrteCfg........:
         Shotput.rtp.raleigh.ibm.com

   Authorization server host used to run SrvSslCfg.:
         Shotput.rtp.raleigh.ibm.com
```

Figure 7-44 Configure external authorization - Policy server settings (1 of 2)

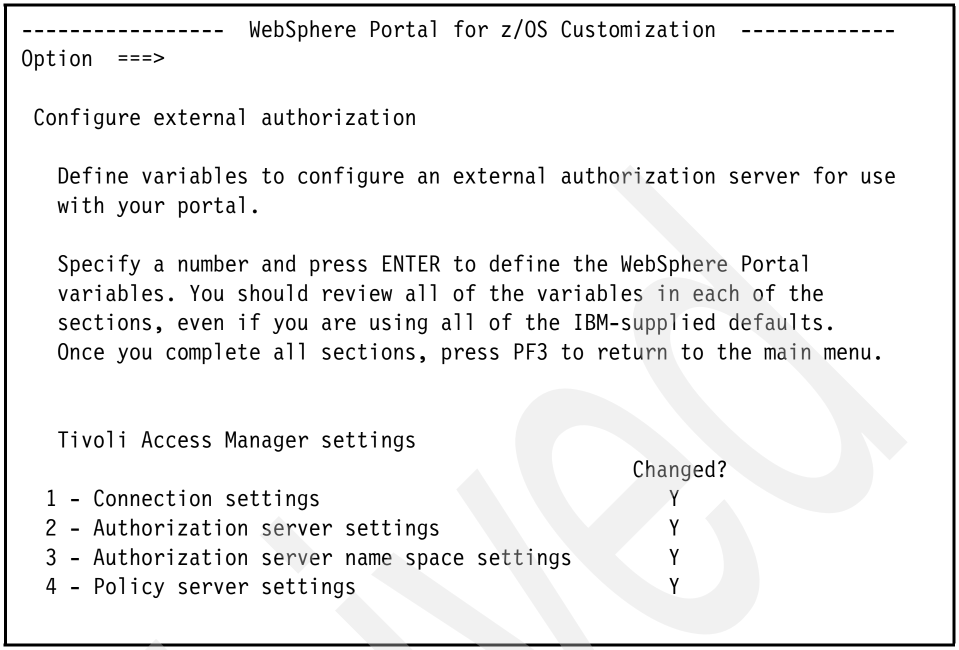
Figure 7-45 on page 291 displays the second part of the Policy server settings.



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

 Configure external authorization
   Policy server settings (2 of 2)

   Specify the following for the system on which you are installing
   WebSphere Portal. Then press ENTER to continue.

   Policy servers used to run SrvSslCfg.:

   Server 1.:   Shotput.rtp.raleigh.ibm.com:7135:1
   Server 2.:
   Server 3.:
   Server 4.:

   Authorization servers................:

   Server 1.:   Shotput.rtp.raleigh.ibm.com:7136:1
   Server 2.:
   Server 3.:
   Server 4.:

   Key path.:
    /wasndconfig/ndcell/ndnode/AppServer/java/jre/pdperm.ks
```

Figure 7-45 Configure external authorization - Policy server settings (2 of 2)

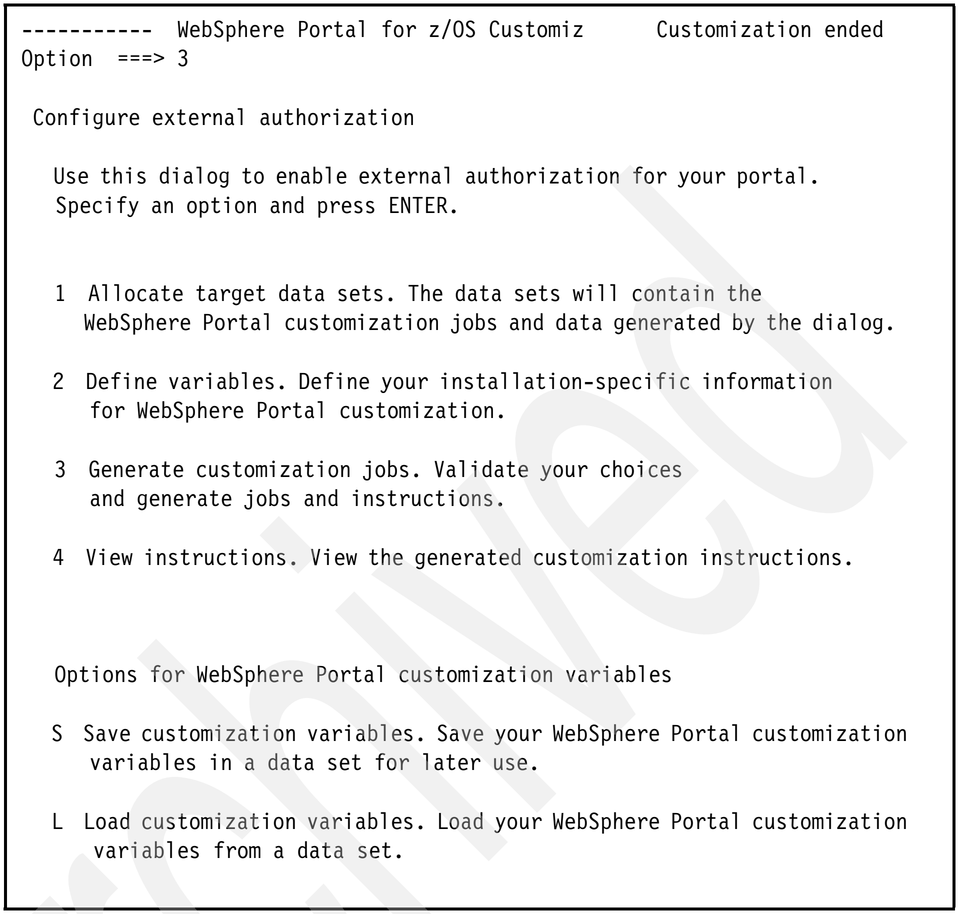
29. Use F3 to return to the main menu; this saves the variables that you just defined. See Figure 7-46 on page 292.



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

 Configure external authorization

   Define variables to configure an external authorization server for use
   with your portal.

   Specify a number and press ENTER to define the WebSphere Portal
   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.


   Tivoli Access Manager settings
                                                 Changed?
  1 - Connection settings                           Y
  2 - Authorization server settings                 Y
  3 - Authorization server name space settings      Y
  4 - Policy server settings                        Y
```

Figure 7-46 Returning to the main menu

30. Receiving the panel shown in Figure 7-47 on page 293 indicates you have finished defining the variables. The message `Customization ended` indicates you are ready to generate the customization jobs.

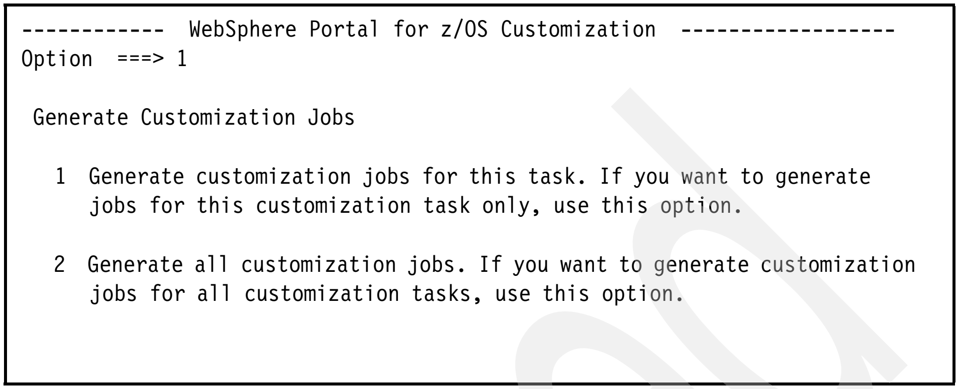


```
-----------  WebSphere Portal for z/OS Customiz      Customization ended
Option  ===> 3

 Configure external authorization

   Use this dialog to enable external authorization for your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.



   Options for WebSphere Portal customization variables

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 7-47 WebSphere Portal for z/OS Customization - Customization ended

At this point, we had completed defining the variables required to configure external authorization, and were ready to generate the JCL and data members for the jobs to configure external authorization.

31. Select option **1 - Generate customization jobs for this task.**, as shown in Figure 7-48.

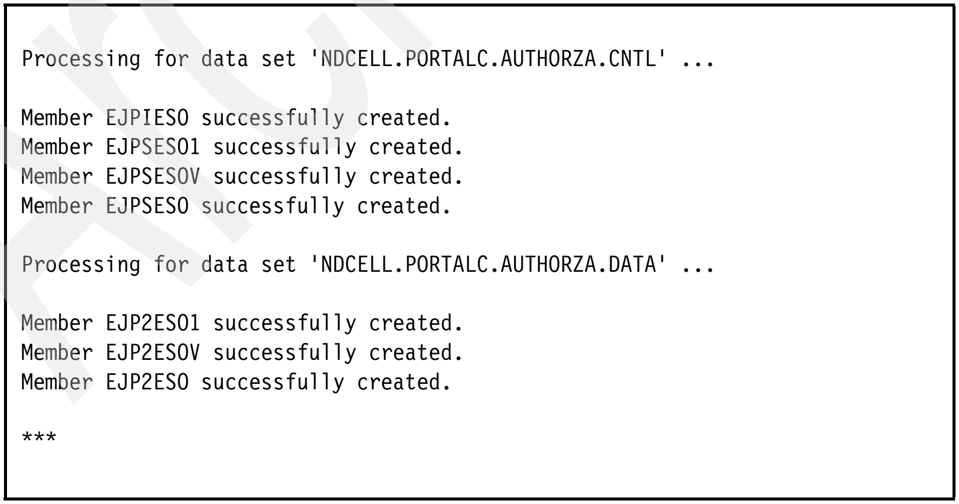


```
------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1

 Generate Customization Jobs

   1  Generate customization jobs for this task. If you want to generate
      jobs for this customization task only, use this option.

   2  Generate all customization jobs. If you want to generate customization
      jobs for all customization tasks, use this option.
```

Figure 7-48 WebSpherePortal for z/OS Customization - Generate Customization Jobs

> **Note:** We recommend that you only generate the customization jobs for this task.
>
> Selecting option 2 will generate *all* customization jobs for *all* customization tasks in the .CNTL and .DATA data sets.

32. The files required to run the configuration jobs are created, as shown in Figure 7-49.



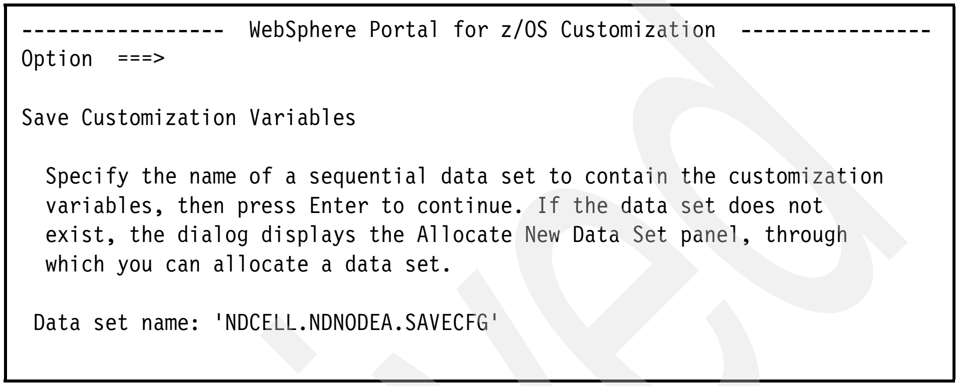
```
Processing for data set 'NDCELL.PORTALC.AUTHORZA.CNTL' ...

Member EJPIESO successfully created.
Member EJPSESO1 successfully created.
Member EJPSESOV successfully created.
Member EJPSESO successfully created.

Processing for data set 'NDCELL.PORTALC.AUTHORZA.DATA' ...

Member EJP2ESO1 successfully created.
Member EJP2ESOV successfully created.
Member EJP2ESO successfully created.

***
```

Figure 7-49 WebSpherePortal for z/OS Customization - Creating the JCL members

33. Press Enter to return to the “Generate customization jobs” panel.

34. Select **S - Save customization variables** and press Enter.

In our case, we saved the environment customization variables in the same data set (`NDCELL.NDNODEA.SAVECFG`) we used in our previous configuration steps; see Figure 7-50.



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

Save Customization Variables

  Specify the name of a sequential data set to contain the customization
  variables, then press Enter to continue. If the data set does not
  exist, the dialog displays the Allocate New Data Set panel, through
  which you can allocate a data set.

 Data set name: 'NDCELL.NDNODEA.SAVECFG'
```

Figure 7-50 WebSphere Portal for z/OS Customization - Save Customization Variables

35. Select option **4 - View the instructions** and press Enter.

Use the instructions in member EJPIESO of the .CNTL dataset to run the jobs required to configure external authorization, as listed in Table 7-10.

Table 7-10 Members created in data set NDCELL.PORTALC.AUTHORZA.CNTL

Member name	Description
EJPIESO	Instructions for customizing WebSphere Portal for z/OS, configure external authorization.
EJPSESO1	WebSphere Portal Server configure PD Policy Server SSL job. This job is used to configure an SSL connection between a Tivoli Access Manager server and WebSphere Portal. If you have not previously run this job, you may run it here.
EJPSESOV	WebSphere Portal Server validate PD connection job. This job is used to verify connectivity to your Tivoli Access Manager server.

Member name	Description
EJPSESO	WebSphere Portal Server enable TAM Authz Server job This job is used to configure a Tivoli Access Manager Authorization server for use with WebSphere Portal.

Table 7-11 lists the members created in NDCELL.PORTALC.AUTHORZA.DATA.

Table 7-11 Members created in data set NDCELL.PORTALC.AUTHORZA.DATA

Member name	Description
EJP2ESO	WebSphere Portal configuration helper file for z/OS for advanced config task - configure TAM authorization server
EJP2ESOV	WebSphere Portal configuration helper file for z/OS for advanced config task - validate PD Admin connection
EJP2ESO1	WebSphere Portal configuration helper file for z/OS for advanced config task - configure Tam Policy Server

You must follow the instructions in NDCELL.PORTALC.AUTHORZA.CNTL(EJPIESO) in order to complete the configuration of external authorization.

Note that you need to bring down the WebSphere Portal before running these jobs. If running in a cluster environment, stop all cluster members. The WebSphere Application Server administration server must be running. After successful completion of the jobs, restart the WebSphere Portal.

In our case, we did not encounter any errors when running the jobs to configure external authorization.

Verifying external authorization using TAM

To verify externalized authorization, follow these steps. Refer to the InfoCenter security topic “Verify that Tivoli Access Manager is working properly” for more detailed information about this topic.

1. Log onto the TAM Administration server. Expand the Object Space and select Browse Object Space. Figure 7-51 shows the expanded Object Space.

   Note that the TAM Authorization server settings we defined in Figure 7-40 on page 286 are shown in the TAM_itso_nd50 object space structure.

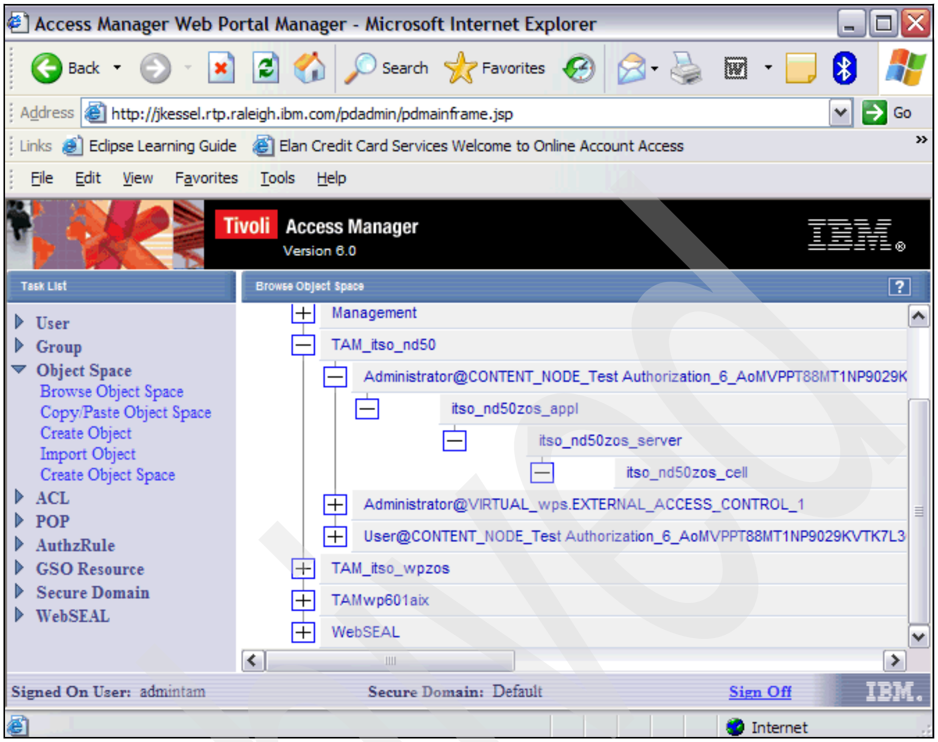
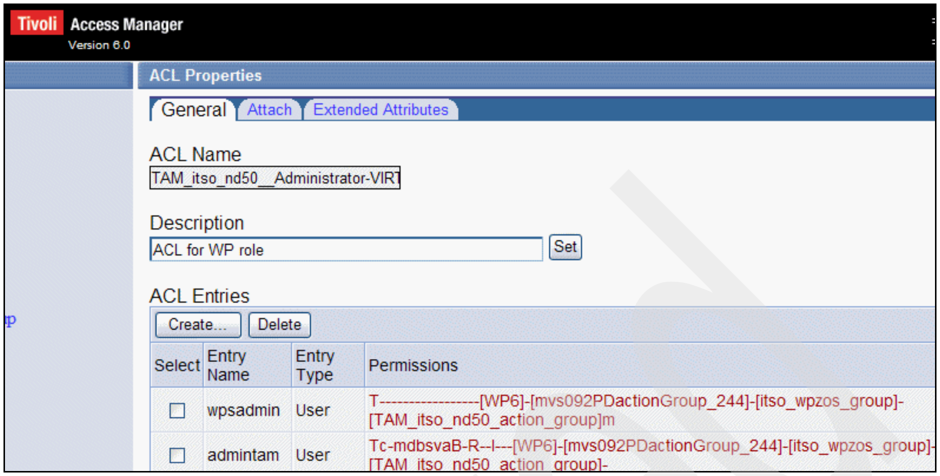


Figure 7-51 TAM Administration server showing TAM_itso_nd50 Object Space

2. Verify that at least one user has the Adminstrator@VIRTUAL/EXTERNAL ACCESS CONTROL_1 role.

   From the TAM Administrator, expand ACL. Select **List ACL** and find TAM_itso_nd50__Administrator-VIRTUAL_wps-EXTERNAL_ACCESS_CONTROL_1.

   Verify that this role has at least one user. In our case, as shown in Figure 7-52 on page 298, there are two users listed.



ACL Name
TAM_itso_nd50__Administrator-VIRT

Description
ACL for WP role

ACL Entries

Select	Entry Name	Entry Type	Permissions
□	wpsadmin	User	T----------------[WP6]-[mvs092PDactionGroup_244]-[itso_wpzos_group]-[TAM_itso_nd50_action_group]m
□	admintam	User	Tc-mdbsvaB-R--l---[WP6]-[mvs092PDactionGroup_244]-[itso_wpzos_group]-[TAM_itso_nd50_action_group]-

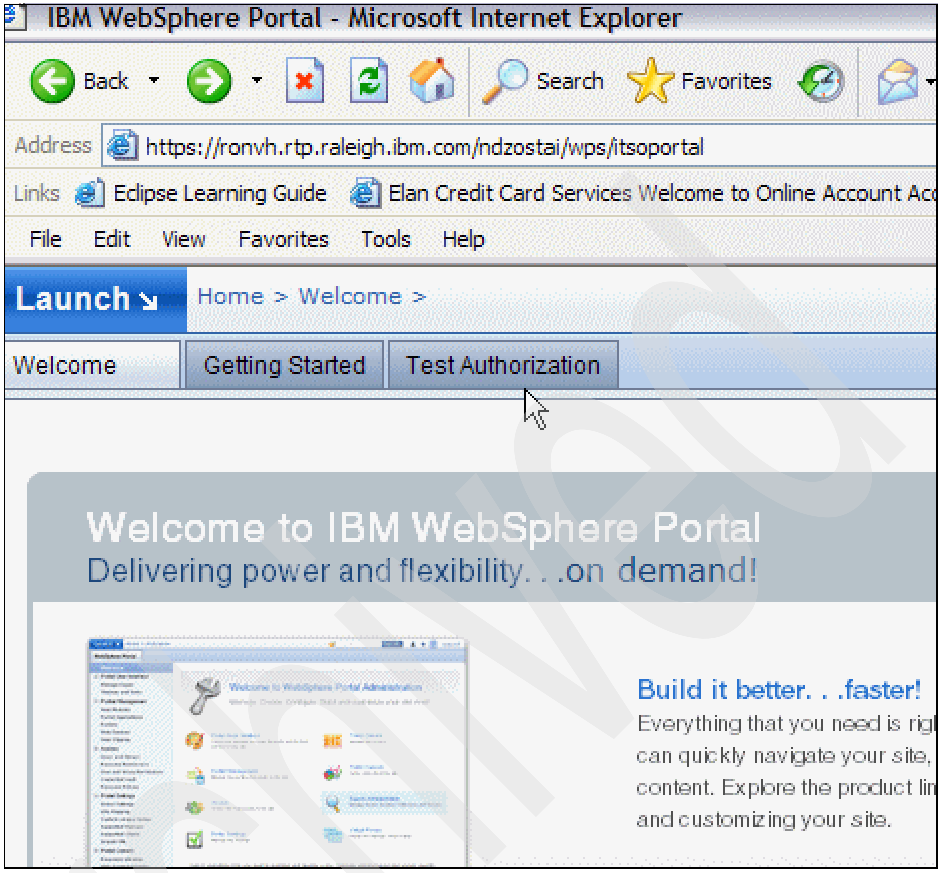
Figure 7-52 TAM ACL - TAM_itso_nd50__Administrator-VIRTUAL_wps-EXTERNAL_ACCESS_CONTROL_1

If there is no entry for the Portal administrator, use the pdadmin command to add the Portal administrator. From the pdadmin command line, enter the following command, where wpsadmin is the Portal administrator user ID and wpsadmins is the portal administrator group:

```
pdadmin> acl modify
WPS_Administrator-VIRTUAL_wps-EXTERNAL_ACCESS_CONTROL_1 set user
wpsadmin T[WPS]m

pdadmin> acl modify
WPS_Administrator-VIRTUAL_wps-EXTERNAL_ACCESS_CONTROL_1 set group
wpsadmins T[WPS]m
```

3. To test that we could now externalize a resource, in our case we logged on to the WebSpherePortal administrator and created a new page, called Test Authorization.

   We used the Resource Permissions portlet and selected **Portal user Interface** > **Manages Pages** > **Select page** >**Content Root** and used the New Page wizard. We assigned the name Test Authorization to this new page; see Figure 7-53 on page 299.


IBM WebSphere Portal - Microsoft Internet Explorer
Back
Search
Favorites
Address
https://ronvh.rtp.raleigh.ibm.com/ndzostai/wps/itsoportal
Links
Eclipse Learning Guide
Elan Credit Card Services Welcome to Online Account Acc
File
Edit
View
Favorites
Tools
Help
Launch ↘
Home > Welcome >
Welcome
Getting Started
Test Authorization

Welcome to IBM WebSphere Portal

Delivering power and flexibility. . .on demand!

Build it better. . .faster!

Everything that you need is rig
can quickly navigate your site,
content. Explore the product lin
and customizing your site.

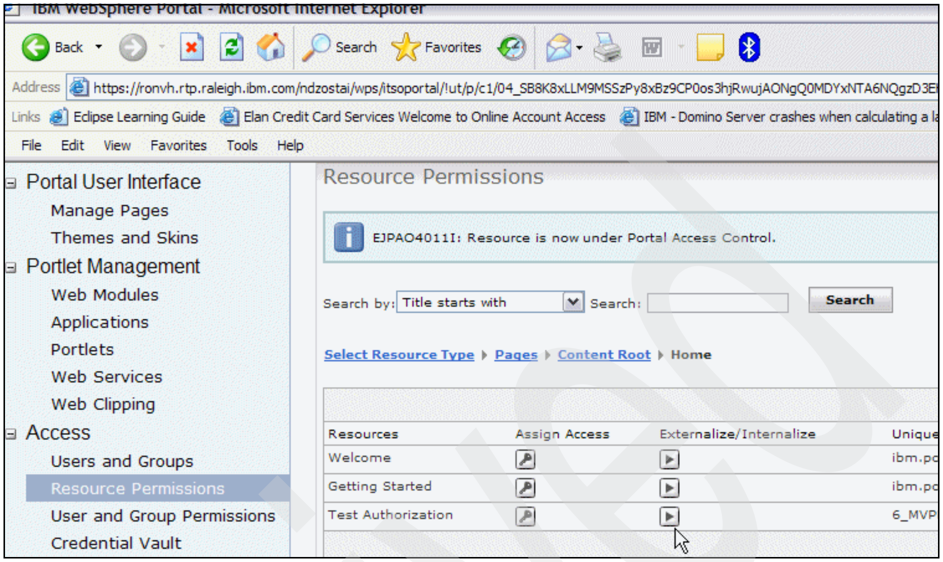
Figure 7-53 Test Authorization Page - seen from user ID wpsadmin

Initially this new resource is assigned default access and is managed internally. The group All Authenticated Users is assigned access to this resource by default. We had to change this to block access for the group All Authenticated Users from this resource and then externalize and test it.

Still using the Resource Permissions portlet, to externalize the authorization for this page we selected **Access > Resource Permissions > Pages > Content Root > Home**. This displays the resources under the Home page.

The Assign Access and Externalize/Internalize buttons are shown in Figure 7-54 on page 300. For more detailed information about using the Resource Permissions portlet to administer WebSphere Portal resources, refer to “Take a Test Drive of the Site” in the InfoCenter:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wps/gs.htm`



IBM WebSphere Portal - Microsoft Internet Explorer

https://ronvh.rtp.raleigh.ibm.com/ndzostai/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjRwujAONgQ0MDYxNTA6NQgzD3E

Eclipse Learning Guide | Elan Credit Card Services Welcome to Online Account Access | IBM - Domino Server crashes when calculating a la

- Portal User Interface
  - Manage Pages
  - Themes and Skins
- Portlet Management
  - Web Modules
  - Applications
  - Portlets
  - Web Services
  - Web Clipping
- Access
  - Users and Groups
  - Resource Permissions
  - User and Group Permissions
  - Credential Vault

Resource Permissions

EJPAO4011I: Resource is now under Portal Access Control.

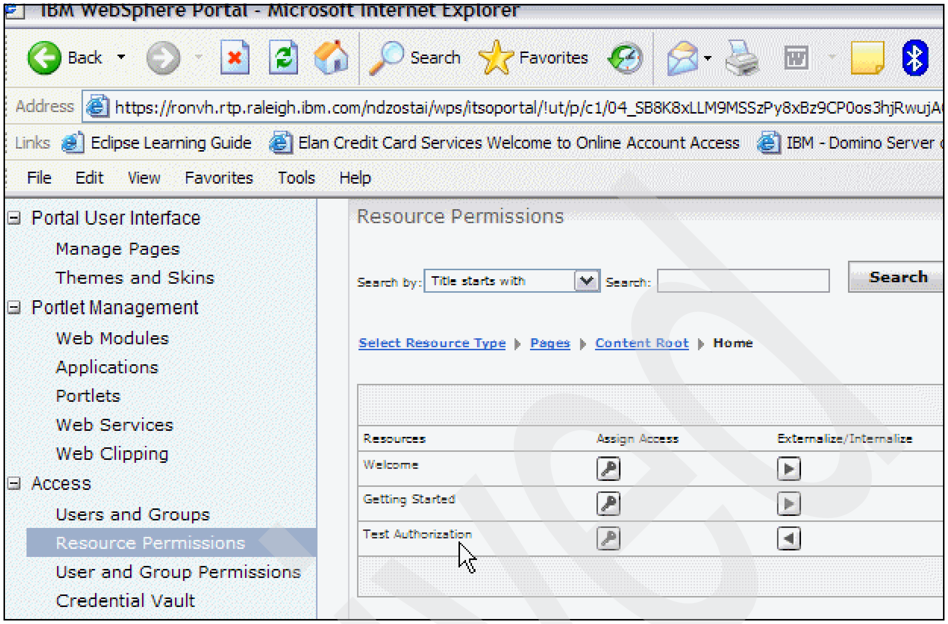
Search by: Title starts with Search: Search

Select Resource Type ▸ Pages ▸ Content Root ▸ Home

Resources	Assign Access	Externalize/Internalize	Unique
Welcome	🔑	▶	ibm.pc
Getting Started	🔑	▶	ibm.pc
Test Authorization	🔑	▶	6_MVP

Figure 7-54 Page Test Authorization is internalized

After a resource is externalized, all inheritance is blocked. To test this, we first tested accessing the resource while it was still internalized. Using two different authenticated users, we verified that this new page was available from the Home page at login time; see Figure 7-55 on page 301.



https://ronvh.rtp.raleigh.ibm.com/ndzostai/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjRwujA

Resource Permissions

Search by: Title starts with Search:

Select Resource Type ▸ Pages ▸ Content Root ▸ Home

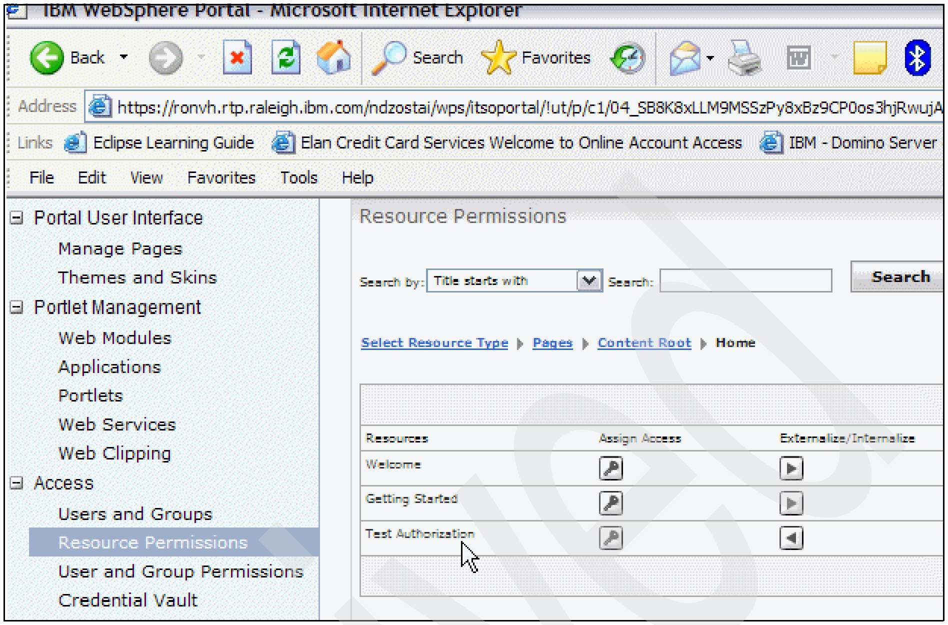
Resources	Assign Access	Externalize/Internalize
Welcome	🔑	▶
Getting Started	🔑	▶
Test Authorization	🔑	◀

Figure 7-55 Page Test Authorization is externalized

Then we externalized the Test Authorization page, using the Resource Permissions portlet. By clicking the Internalize/Externalize button, we externalized the Test Authorization page.

Now the WebSphere Portal administrator can no longer administer this resource. Instead, it must be done through the external TAM administrator; see Figure 7-56 on page 302.

However, the parent page (Home) is still internalized and administered by the WebSphere Portal administrator.



Resources	Assign Access	Externalize/Internalize
Welcome	🔑	▶
Getting Started	🔑	▶
Test Authorization	🔑	◀

Figure 7-56 Page Test Authorization externalized by WebSphere Portal admin

After the page has been externalized, it no longer inherits access from the parent document. The Home page is not restricted, so any child resources that are internalized inherit this access level.

However, after we externalized page Test Authorization, *this* inheritance was broken—and now this resource can only be managed by TAM. The only action WebSphere Portal admin can do is to internalize it to move it back under WebSphere Portal admin control.

8

Configuring Web Content Management

In this chapter we show how to install and configure the Web Content Management (WCM) component of IBM WebSphere Portal Enable for z/OS Version 6.0.0.1.

We cover these topics in the following sections:

- Introduction to IBM Workplace Web Content Management
- Configuring DCS and Spell Checker on the remote server
- WCM configuration on the Portal on z/OS

8.1 Introduction to IBM Workplace Web Content Management

IBM Workplace Web Content Management (WCM) is a Web content management solution designed to simplify and accelerate development and delivery of critical business information across an enterprise. It enables end-to-end collaboration for content creation, approvals, management, retention, and publishing across Portal, Internet, intranet, and extranet assets.

WCM provides a collaborative environment that make it easy for users of all levels in the business process to work together to complete review and approval workflow processes. Content can be created using a "what you see is what you get" (WYSIWYG) rich text editor, or text can be imported from other applications. Versioning and rollback can be applied to content.

WCM can be used in conjunction with other Portal features such as Personalization and Portal Document Manager (PDM). WCM users have access to the artifacts in the Portal Document Manager libraries. WCM users can create personalization rules that can be used to deliver content relevant to users and their roles.

WCM separates the design of the Web site from the creation of content via templates. Authoring templates ensure consistency of content. Presentation templates ensure consistency in how that content gets displayed on the site. If the design of the site needs to change, the content is not affected and vice versa.

In the following sections we discuss:

- WCM architecture
- WCM libraries and configuration scenarios

8.2 WCM architecture

WCM is a Portal-based architecture. When Portal is installed, WCM is also installed. In order to access WCM, it must be configured.

WCM is tightly integrated with the Portal and WebSphere Application Server infrastructure, as shown in Figure 8-1.

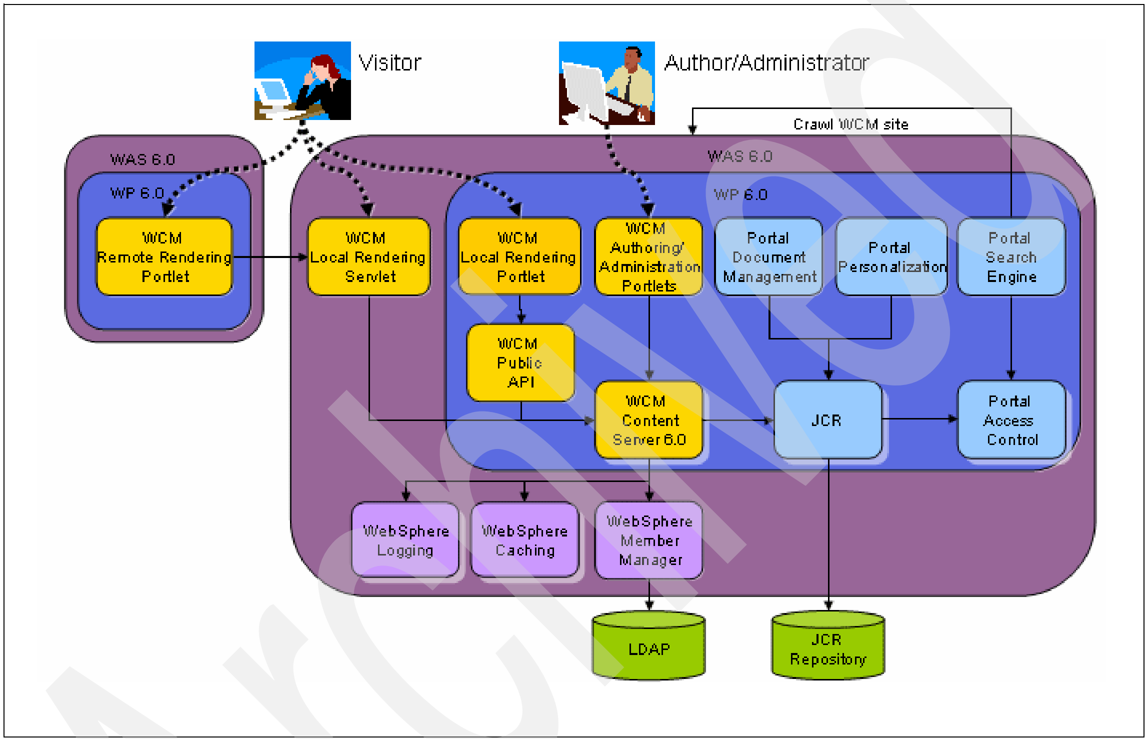


Figure 8-1 WCM architecture

WCM provides various means for content delivery to visitors. WCM portlets deliver content to Portal users. The WCM Local Rendering portlet uses the WCM API to access the WCM content server to display Web content that is located on the same server as WCM. The Remote Rendering portlet connects directly to the WCM content server to display published content only located on different Portal servers.

WCM servlets deliver content to Web sites. WCM JSP™ tags can be used for integration with existing JSP applications. Pre-rendered HTML can be used for static Web sites. By using the WCM API, WCM users can create custom applications.

WCM authors and administrators use the WCM Authoring and Administration portlet to contribute content and administer WCM, respectively. For more information about this portlet, refer to the WebSphere Portal infocenter article entitled “Using the authoring portlet”, which is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wcm/wcm_dev_using.html

WCM, Personalization, and Portal Document Manager all use the Java Content Repository (JCR) database. This JCR database can be configured to integrate with any JCR supported repository, such as DB2 Content Manager.

WCM has access to the users in groups in the LDAP registry because it relies on WebSphere Application Server for security. Portal controls the access (authorization) to the artifacts in the JCR database. The Portal infocenter provides an access control example in an article entitled “Library access control example”, which is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wcm/wcm_security_library_example.html

8.2.1 WCM libraries and configuration scenarios

All WCM content is stored in a JCR database as *libraries*. Libraries are a new data partitioning feature in WCM. There is a default WCM library created when Portal is installed called Web Content.

The use of multiple libraries allows for separation of data. All WCM artifacts for one site can be kept in its own library. The advantages of using libraries include:

- Each site can have its own library.
- Separate Presentation content from Authoring content.
- Isolate test content.

Data gets replicated from one WCM library into another WCM library via a process called *syndication*. Syndication requires a *syndicator* and subscriber to be defined. The syndicator defines a connection to a subscriber and indicates which libraries to syndicate to the subscriber. The subscriber defines a connection to the syndicator and receives the data replicated from the libraries specified by the syndicator. Syndicators and subscribers can have a one-way or two-way relationship.

> **Note:** If using a two-way syndication, you must first establish the syndication relationship from syndicator to subscriber. Then you can establish a relationship between subscriber and syndicator.

Tip: Syndication to an existing library with the same name is not supported, so create a new WCM library before creating any WCM artifacts.

The Portal Infocenter provides more information about libraries and syndication in an article entitled “Syndication”, which is available at the following site:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wcm/wcm_syndication.html`

There are three common WCM configuration scenarios:

- WCM can be configured as an authoring server.

  In this scenario, WCM is used to create and manage Web content.
- WCM can be configured as a *staging application*.

  In this scenario, WCM is an interim server that might contain aggregate changes to Web content before they are published to production.
- WCM can be configured as a *delivery server*.

  In this scenario, WCM is used to display Web content to end users.

These servers can be clustered so there will be no single point of failure. The Portal infocenter provides more detail about how to configure each of these scenarios in an article entitled “Configuration Scenarios”, which is available at the following site:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v60r0/topic/com.ibm.wp.zos.doc/wcm/wcm_install_scenarios.html`

Figure 8-2 illustrates an example of syndication across multiples libraries on various WCM servers.

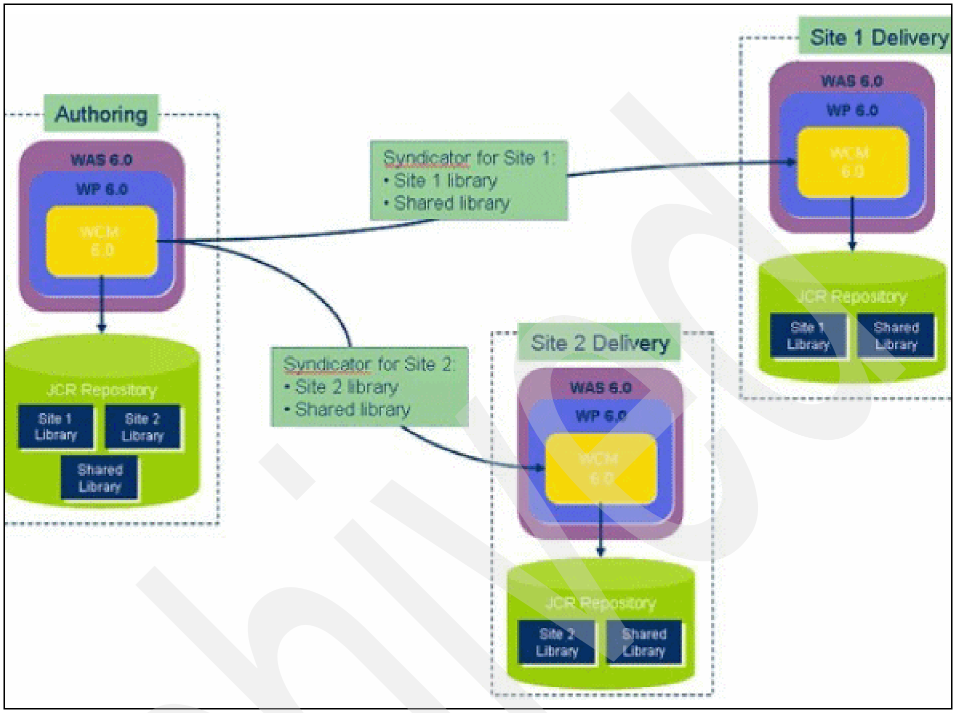


Figure 8-2 Libraries and syndication

8.2.2 WCM terminology

Table 8-1 lists WCM terminology that is important to understand before you work with WCM.

Table 8-1 WCM terminology

Concept	Description
Elements	They either store Web content, or generate Web content and are stored in items. Elements are also referred to as Content Components.

Concept	Description
Content Components	They store elements. There are four types: container; template; profile; workflow. Content Components are also referred to as Items.
Container items	They represent different sections of a site.
Template items	They are used to separate the creation of items (authoring) from the design of the site (presentation).
Authoring templates	They define which data entry fields are visible on a content item form, the default values for each setting and field on a content item form.
Presentation templates	They define the layout of the elements and components that are displayed on a page, as well as the default properties of a page.
Profile items	They define taxonomies and categories of content item types. Categories are used to classify content and are grouped with taxonomies.
Workflow items	They are used to control the access to, verification, and eventual approval of WCM items.
Workflow stages	They determine what actions to execute when entering or exiting a workflow stage, and the access levels of users or groups within that stage.
Workflow actions	They are executed when entering or exiting a workflow stage.
Site	This is the logical separation of a single Web site on WCM. It typically contains multiple site areas.
Site framework	This the primary content hierarchy for navigating the site. It is similar to a site map, and consists of sites and site areas.
Site area	This is a container for content, other site areas, or both.

Concept	Description
Template maps	They are the pairing of an authoring template with a presentation template. This gets defined in a site or a site area.

8.3 Configuring DCS and Spell Checker on the remote server

In IBM WebSphere Portal Enable for z/OS Version 6.0.0.1, Document Conversion Services (DCS) including Spell Checker must be performed on a separate distributed server (remote DCS server) because these services are not supported within z/OS at this release. The remote DCS server can be implemented on a Windows-based, Unix-based or Linux-based platform. In our case, we use a Windows-based server running Windows XP Professional Edition.

You will probably want to allow users to search content. We do not describe in detail how to configure the search engine, but you should note that when Portal is configured in a cluster, the search engine must be deployed in a separate WebSphere Application Server. That server could be running on z/OS, but it cannot be clustered. Refer to the WebSphere Portal Infocenter article “Configuring search in a clustered environment” for details.

The remainder of this chapter guides you through the steps to install and configure these services.

Configuration of the WCM components of WebSphere Portal on z/OS consists of the following steps, which must be performed in the sequence shown on the remote DCS server.

- Locate remote DCS and Spell Checker components
- Deploy the dcs.war file
- Deploy the SpellChecker.ear file
- Restart WebSphere application Server on the remote DCS server

In the following sections, we explain these steps in more detail.

8.3.1 Locate remote DCS and Spell Checker components

Initially you need to locate the components that you will need to install on the remote DCS server.

Tip: A simple approach is to copy the files from their source location to an interim location of your choice on your local workstation. Then, as a second step, copy them again to the target location on remote DCS server.

There are five components you need to find. We list these components and their source and target locations in Table 8-2.

Note: Ensure that you use a *binary* file transfer to move the files between z/OS, your local workstation, and the remote DCS server.

Table 8-2 Location information for required remote DCS components

File name	Source location	Target location on remote DCS server
remoteDCS.zip	For IBM WebSphere Portal Enable for z/OS Version 6.0.0.1, download this file from the url: `http://www-1.ibm.com/support/docview.wss?rs=688&uid=swg24014059`	See detail in 8.3.2, “Deploy the dcs.war file” on page 312
SpellChecker.ear	<PortalRoot>/odc/spellchecker/	See detail in 8.3.3, “Deploy the SpellChecker.ear file” on page 323
spcommon.jar	<PortalRoot>/odc/spellchecker/lib	<AppserverRoot>/lib/ext
SpellCheckConfig.properties	<PortalRoot>/odc/spellchecker/lib/com/ibm/wps/odc/spellcheck/util/	<AppserverRoot>/lib/ext/com/ibm/wps/odc/spellcheck/util
log_sp.properties	<PortalRoot>/odc/spellchecker/lib/com/ibm/wps/odc/spellcheck/util/	<AppserverRoot>/lib/ext/com/ibm/wps/odc/spellcheck/util

In our case, the <PortalRoot> directory on z/OS is /usr/lpp/zPortalServerWP/V6R0.

On our remote DCS server, the <AppserverRoot> directory is C:\WebSphere\AppServer\profiles\z707_win_dcs1\installedApps.

If you determine that these files are not in the source location you expect, you can use the UNIX System Services (USS) **`find`** command in an OMVS session to locate the files in the Hierarchical File System (HFS), as shown in Example 8-1 on page 312.

Example 8-1 Finding the required components in the HFS

```
DAVIES @ SC49:/usr/lpp/zPortalServerWP/V6R0>find . -name spcommon.jar
./odc/spellchecker/lib/spcommon.jar
./shared/app/spcommon.jar
```

For the following files (listed from Table 8-2 on page 311), simply copy them to the specified target location on the remote DCS server, remembering to use a *binary* ftp.

- spcommon.jar
- SpellCheckConfig.properties
- log_sp.properties

> **Note:** If the remainder of the directory path <AppserverRoot>/lib/ext/*com/ibm/wps/odc/spellcheck/util* does not already exist on the remote DCS server, create it before copying the files.

8.3.2 Deploy the dcs.war file

After downloading the remoteDCS.zip file from the address shown in Table 8-2 on page 311, you need to extract the contents to a temporary directory on a drive which is accessible to the remote DCS server.

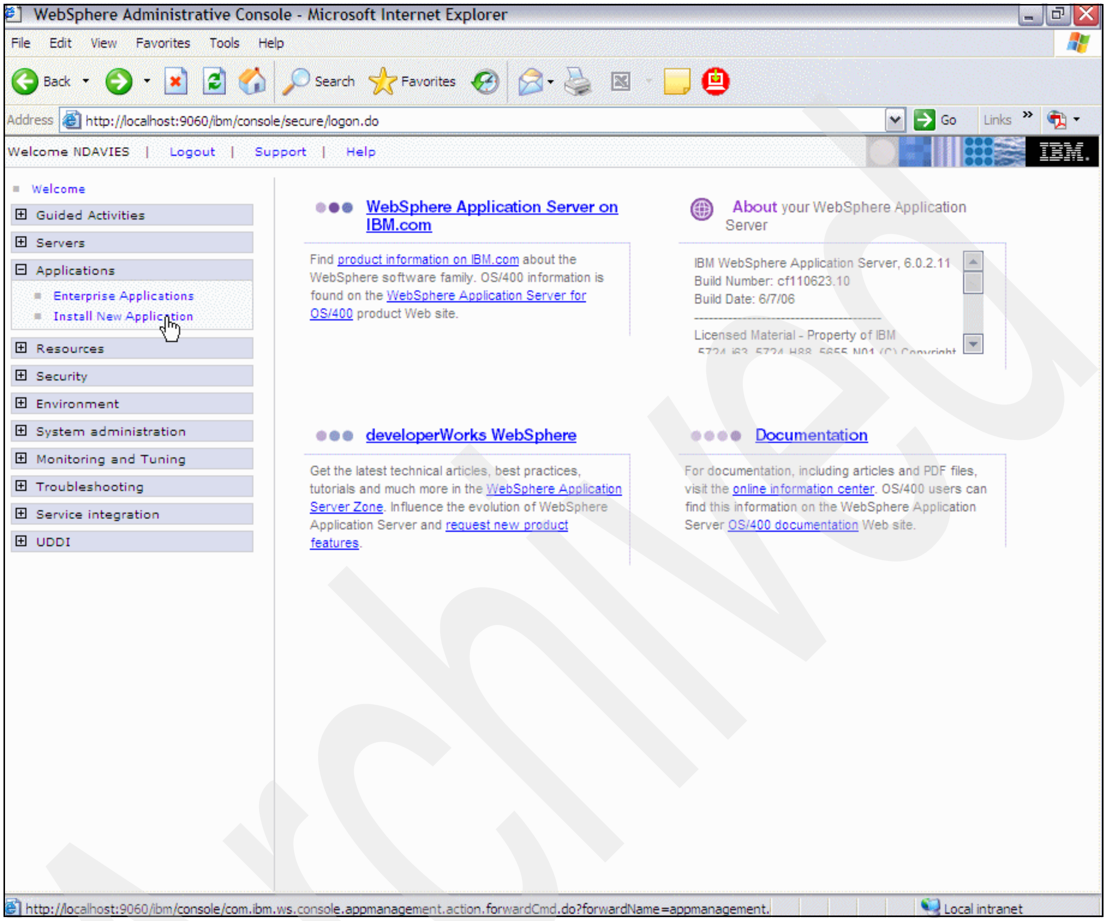
> **Note:** You may find that you cannot see the contents of the remoteDCS.zip file using WinZIP. In that case, use another archiving tool such as PKZip or WinRAR.

When you extract the zip file, it produces a directory called “dcs” with a number of files and subdirectories. You need to locate the file dcs.war in the dcs directory.

Start the WebSphere Application Server Administrative Console on the remote DCS server.

Now follow the steps shown from Figure 8-3 on page 313 to Figure 8-12 on page 322 to deploy the dcs.war file.

Start by clicking **Install New Application** in the WebSphere Application Server Administrative Console, as shown in Figure 8-3.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:9060/ibm/console/secure/logon.do
Go
Links
Welcome NDAVIES | Logout | Support | Help
IBM.
Welcome
Guided Activities
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
http://localhost:9060/ibm/console/com.ibm.ws.console.appmanagement.action.forwardCmd.do?forwardName=appmanagement.
Local intranet

WebSphere Application Server on IBM.com

Find product information on IBM.com about the WebSphere software family. OS/400 information is found on the WebSphere Application Server for OS/400 product Web site.

About your WebSphere Application Server

IBM WebSphere Application Server, 6.0.2.11
Build Number: cf110623.10
Build Date: 6/7/06

Licensed Material - Property of IBM
5724-I63, 5724-H88, 5655-N01 (C) Copyright

developerWorks WebSphere

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

Documentation

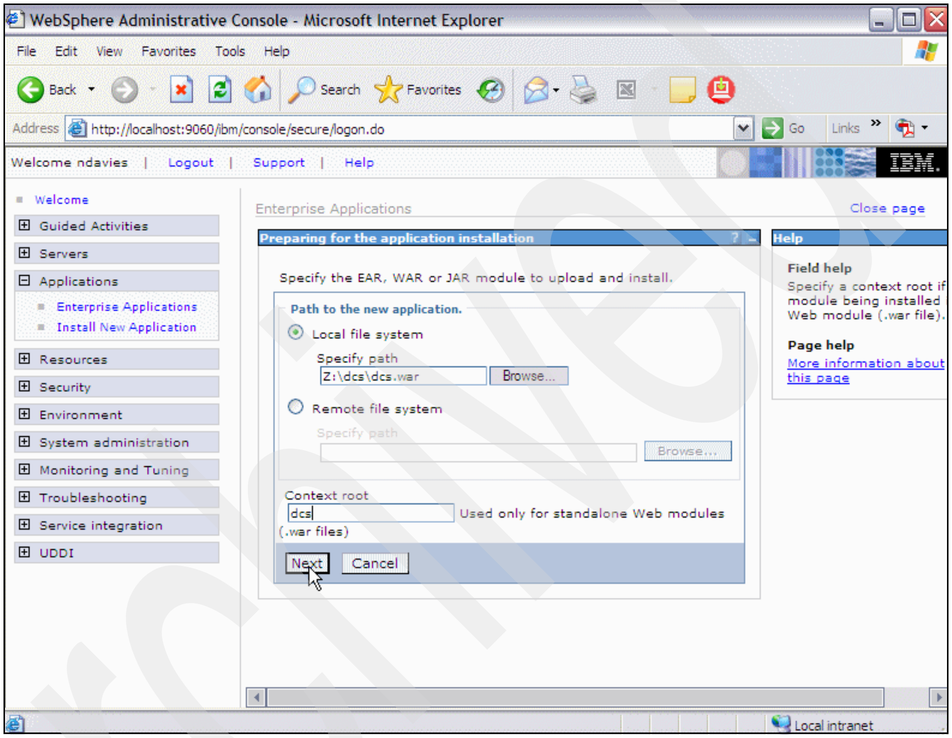
For documentation, including articles and PDF files, visit the online information center. OS/400 users can find this information on the WebSphere Application Server OS/400 documentation Web site.

Figure 8-3 Installing dcs.war file - Step 1

On the screen Specify the EAR, WAR or JAR module to upload and install shown in Figure 8-4 on page 314, you now need to complete the **Specify path** entry field with the path to the dcs.war file which you are going to deploy.

> **Tip:** You can use the **Browse** button under **Local file system** to search for the dcs.war file instead of typing it, if desired.

You also need to specify the **Context root** as dcs and then click **Next**, as shown in Figure 8-4.



Preparing for the application installation

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

● Local file system

Specify path: Z:\dcs\dcs.war [Browse...]

○ Remote file system

Specify path: [Browse...]

Context root: dcs Used only for standalone Web modules (.war files)

[Next] [Cancel]

Help

Field help

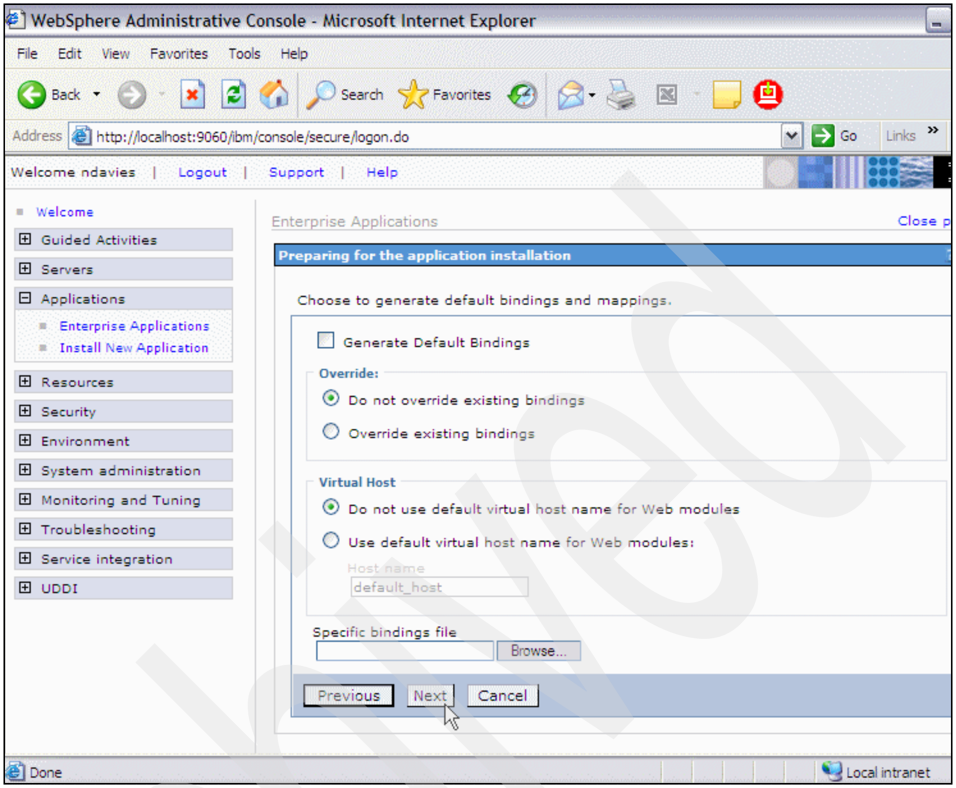
Specify a context root if module being installed Web module (.war file).

Page help

More information about this page

Figure 8-4 Installing dcs.war file - Step 2

Now accept the defaults for existing bindings by clicking **Next**, as shown in Figure 8-5 on page 315.



Preparing for the application installation

Choose to generate default bindings and mappings.

□ Generate Default Bindings

Override:

● Do not override existing bindings

○ Override existing bindings

Virtual Host

● Do not use default virtual host name for Web modules

○ Use default virtual host name for Web modules:

Host name

default_host

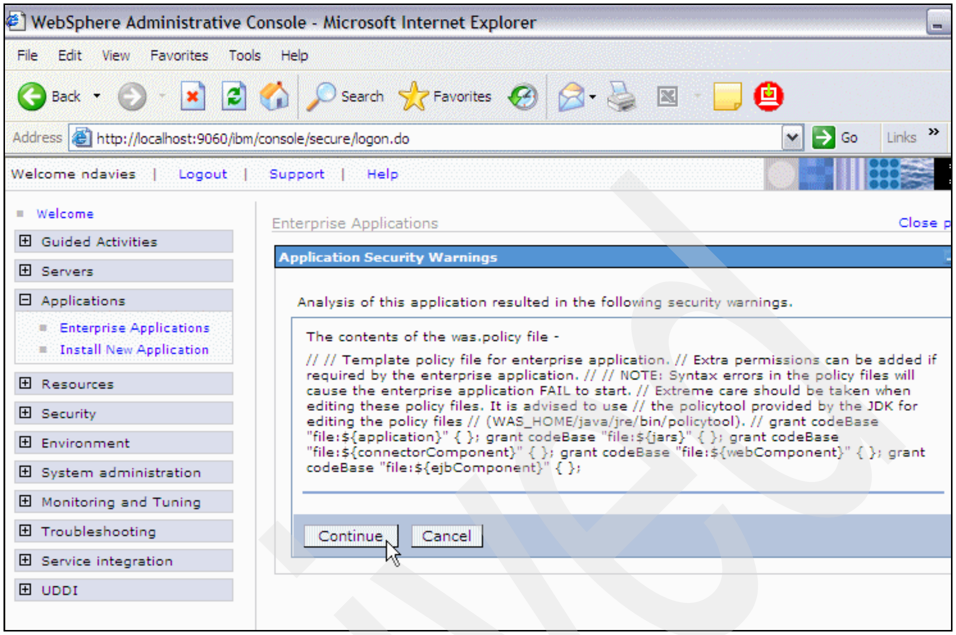
Specific bindings file

Browse...

Previous Next Cancel

Figure 8-5 Installing dcs.war file - Step 3

If you receive an Application Security Warning, accept the security warning by clicking **Continue**, as shown in Figure 8-6 on page 316.



Application Security Warnings

Analysis of this application resulted in the following security warnings.

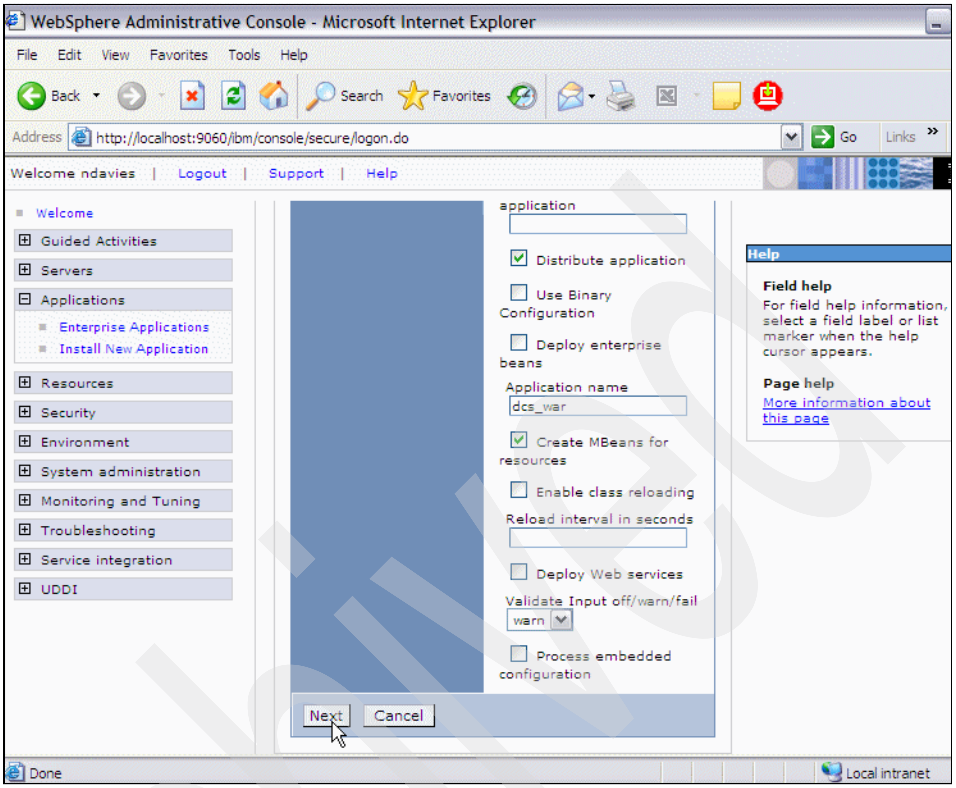
The contents of the was.policy file -

// // Template policy file for enterprise application. // Extra permissions can be added if required by the enterprise application. // // NOTE: Syntax errors in the policy files will cause the enterprise application FAIL to start. // Extreme care should be taken when editing these policy files. It is advised to use // the policytool provided by the JDK for editing the policy files // (WAS_HOME/java/jre/bin/policytool). // grant codeBase "file:${application}" { }; grant codeBase "file:${jars}" { }; grant codeBase "file:${connectorComponent}" { }; grant codeBase "file:${webComponent}" { }; grant codeBase "file:${ejbComponent}" { };

Figure 8-6 Installing dcs.war file - Step 4

Accept the defaults on the screen Select installation options by clicking **Next**, as shown in Figure 8-7 on page 317.

> **Note:** Do *not* change the application name: dcs_war.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:9060/ibm/console/secure/logon.do Go Links
Welcome ndavies | Logout | Support | Help
Welcome
Guided Activities
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Help
Field help
For field help information, select a field label or list marker when the help cursor appears.
Page help
More information about this page
Next Cancel
Done
Local intranet

application

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: dcs_war

☑ Create MBeans for resources

☐ Enable class reloading

Reload interval in seconds:

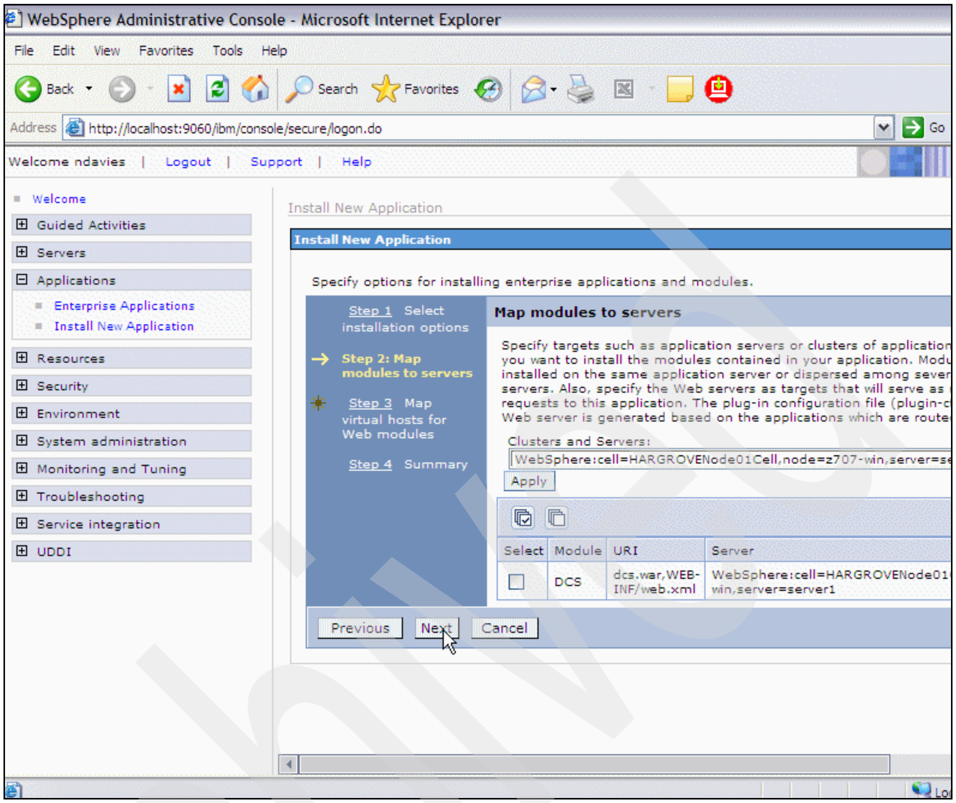
☐ Deploy Web services

Validate Input off/warn/fail: warn

☐ Process embedded configuration

Figure 8-7 Installing dcs.war file - Step 5

Accept the default settings for mapping of modules to servers, as shown in Figure 8-8 on page 318.



http://localhost:9060/ibm/console/secure/logon.do

Install New Application

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

→ **Step 2: Map modules to servers**

Step 3 Map virtual hosts for Web modules

Step 4 Summary

Map modules to servers

Specify targets such as application servers or clusters of application
you want to install the modules contained in your application. Modu
installed on the same application server or dispersed among sever
servers. Also, specify the Web servers as targets that will serve as
requests to this application. The plug-in configuration file (plugin-c
Web server is generated based on the applications which are route

Clusters and Servers:
WebSphere:cell=HARGROVENode01Cell,node=z707-win,server=se

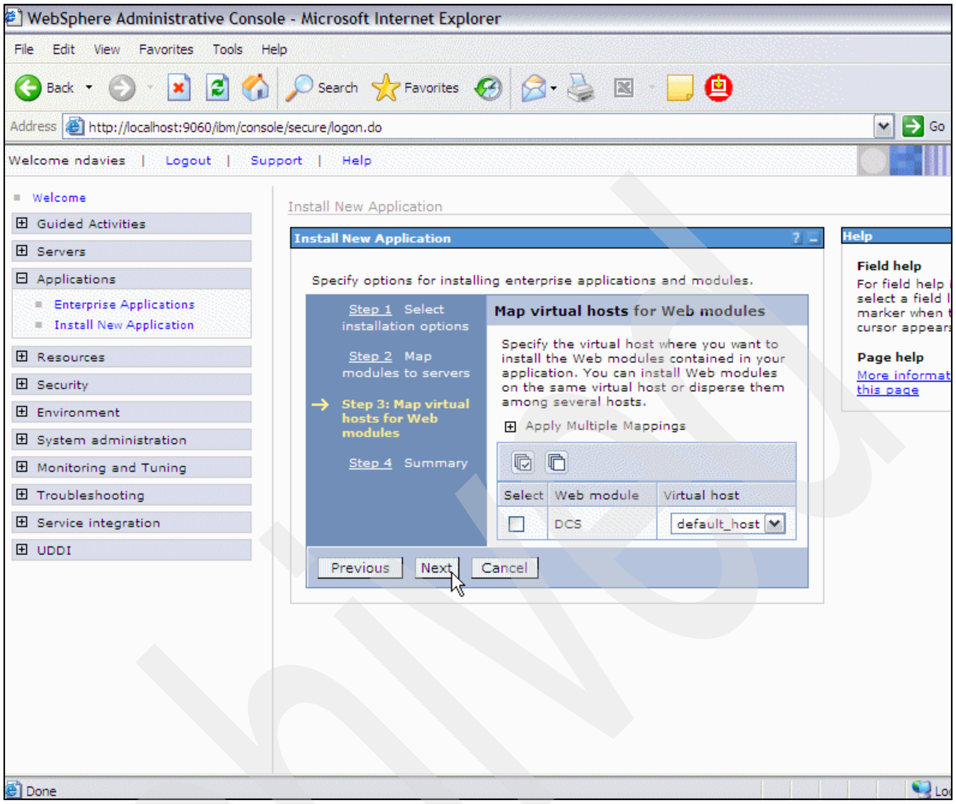
Apply

Select	Module	URI	Server
□	DCS	dcs.war,WEB-INF/web.xml	WebSphere:cell=HARGROVENode01 win,server=server1

Previous Next Cancel

Figure 8-8 Installing dcs.war file - Step 6

Accept the defaults on the screen Map virtual hosts for Web modules by clicking the **Next** button, as shown in Figure 8-9 on page 319.



Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

→ Step 3: Map virtual hosts for Web modules

Step 4 Summary

Map virtual hosts for Web modules

Specify the virtual host where you want to install the Web modules contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

⊞ Apply Multiple Mappings

Select	Web module	Virtual host
□	DCS	default_host

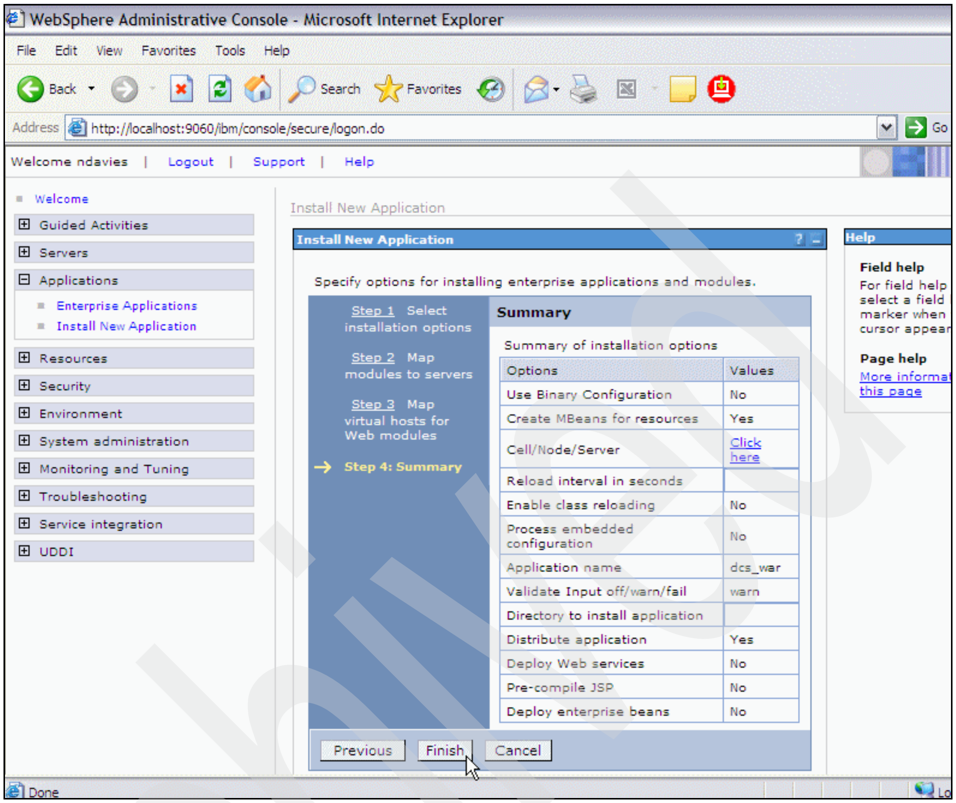
Previous Next Cancel

Field help
For field help select a field marker when cursor appear

Page help
More informat this page

Figure 8-9 Installing dcs.war file - Step 7

On the screen Install new application summary, review your installation choices and proceed by clicking **Finish**, as shown in Figure 8-10 on page 320.



Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map virtual hosts for Web modules

→ Step 4: Summary

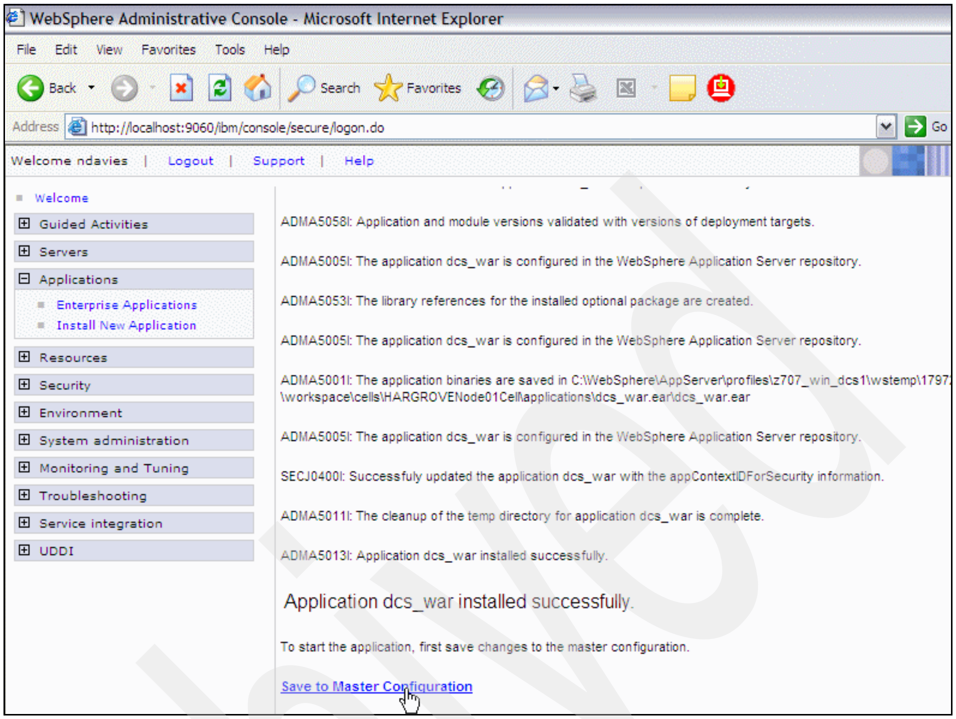
Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	No
Application name	dcs_war
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

Figure 8-10 Installing dcs.war file - Step 8

After the application dcs_war has successfully deployed, click **Save to Master Configuration**, as shown in Figure 8-11 on page 321.



ADMA5058I: Application and module versions validated with versions of deployment targets.

ADMA5005I: The application dcs_war is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application dcs_war is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in C:\WebSphere\AppServer\profiles\z707_win_dcs1\wstemp\1797
\workspace\cells\HARGROVENode01Cell\applications\dcs_war.ear\dcs_war.ear

ADMA5005I: The application dcs_war is configured in the WebSphere Application Server repository.

SECJ0400I: Successfuly updated the application dcs_war with the appContextIDForSecurity information.

ADMA5011I: The cleanup of the temp directory for application dcs_war is complete.

ADMA5013I: Application dcs_war installed successfully.

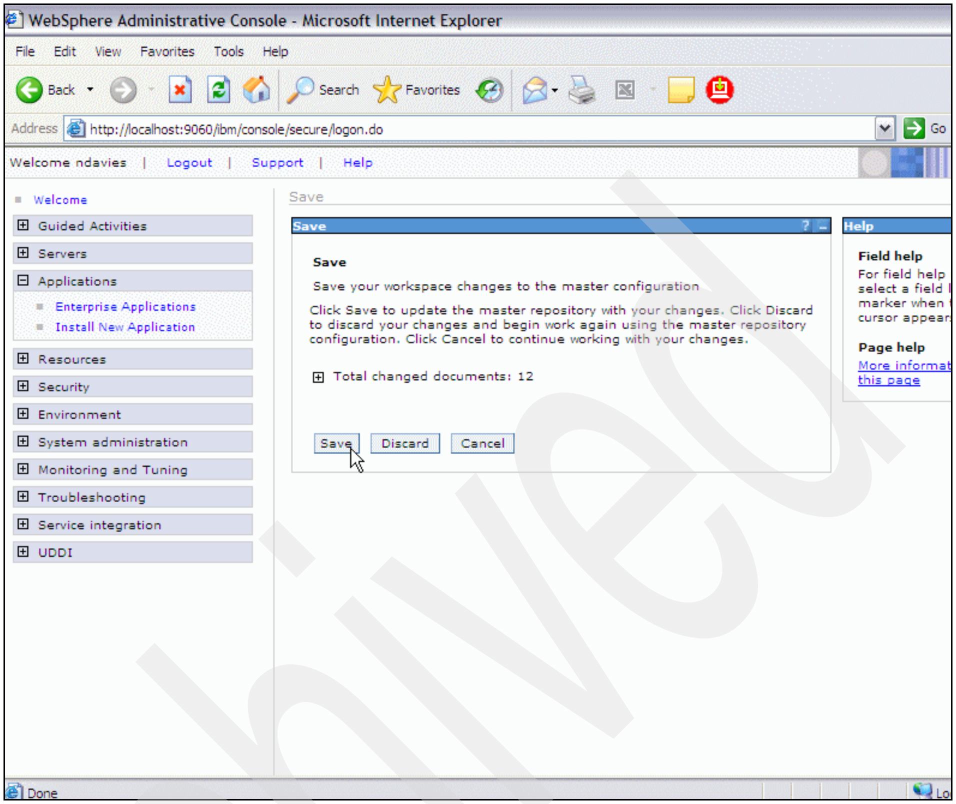
Application dcs_war installed successfully.

To start the application, first save changes to the master configuration.

Save to Master Configuration

Figure 8-11 Installing dcs.war file - Step 9

Finally, complete the application deployment by clicking **Save** to update the WebSphere Application Server master configuration, as shown in Figure 8-12 on page 322 below.



Save

Save your workspace changes to the master configuration

Click Save to update the master repository with your changes. Click Discard to discard your changes and begin work again using the master repository configuration. Click Cancel to continue working with your changes.

⊞ Total changed documents: 12

Field help

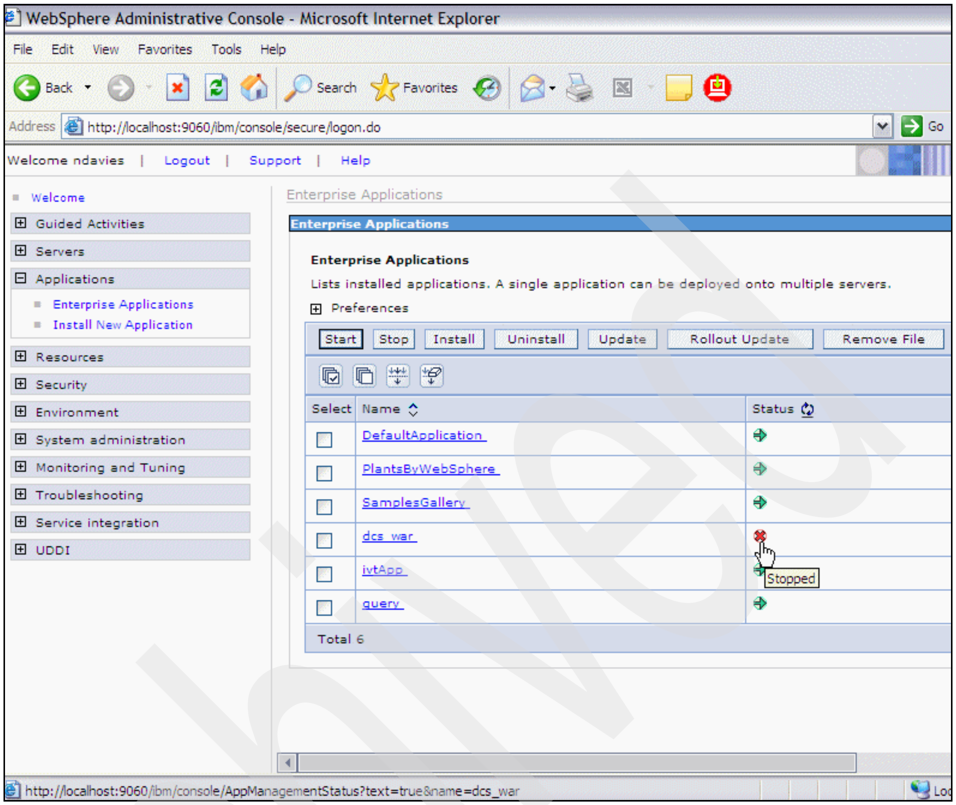
For field help select a field marker when cursor appear

Page help

More informat this page

Figure 8-12 Installing dcs.war file - Step 10

Verify that the `dcs_war` application is now showing in the list of deployed enterprise applications in the WebSphere Application Server Administrative Console and that its status is showing as `Stopped`, as shown in Figure 8-13 on page 323. At this point, allow the application to remain stopped for the time being.



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
□	DefaultApplication	➔
□	PlantsByWebSphere	➔
□	SamplesGallery	➔
□	dcs_war	✖
□	ivtApp	➔
□	query	➔
Total 6		

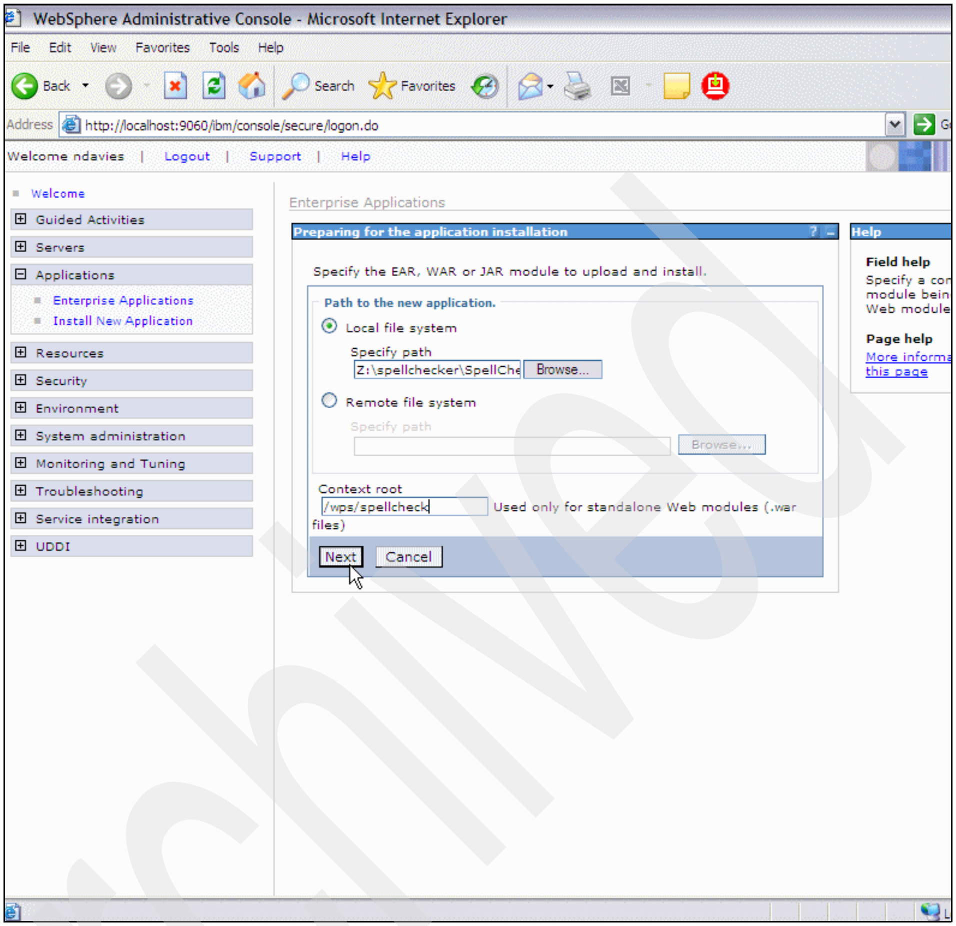
Figure 8-13 Viewing the deployed dcs.war application

8.3.3 Deploy the SpellChecker.ear file

You now need to follow similar steps to those in 8.3.2, “Deploy the dcs.war file” on page 312 to deploy the SpellChecker.ear file on the remote dcs server.

Start by clicking **Install New Application** in the WebSphere Application Server Administrative Console, as shown in Figure 8-3 on page 313.

On the screen Specify the EAR, WAR or JAR module to upload and install shown in Figure 8-14 on page 324, use the **Specify path** field to enter the location of the SpellChecker.ear file in the file system or use the **Browse** button. Specify the context root as `/wps/spellcheck` and click **Next**, as shown in Figure 8-14 on page 324.



Address: http://localhost:9060/ibm/console/secure/logon.do

Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

● Local file system

Specify path: Z:\spellchecker\SpellChe Browse...

○ Remote file system

Specify path: Browse...

Context root: /wps/spellcheck Used only for standalone Web modules (.war files)

Next Cancel

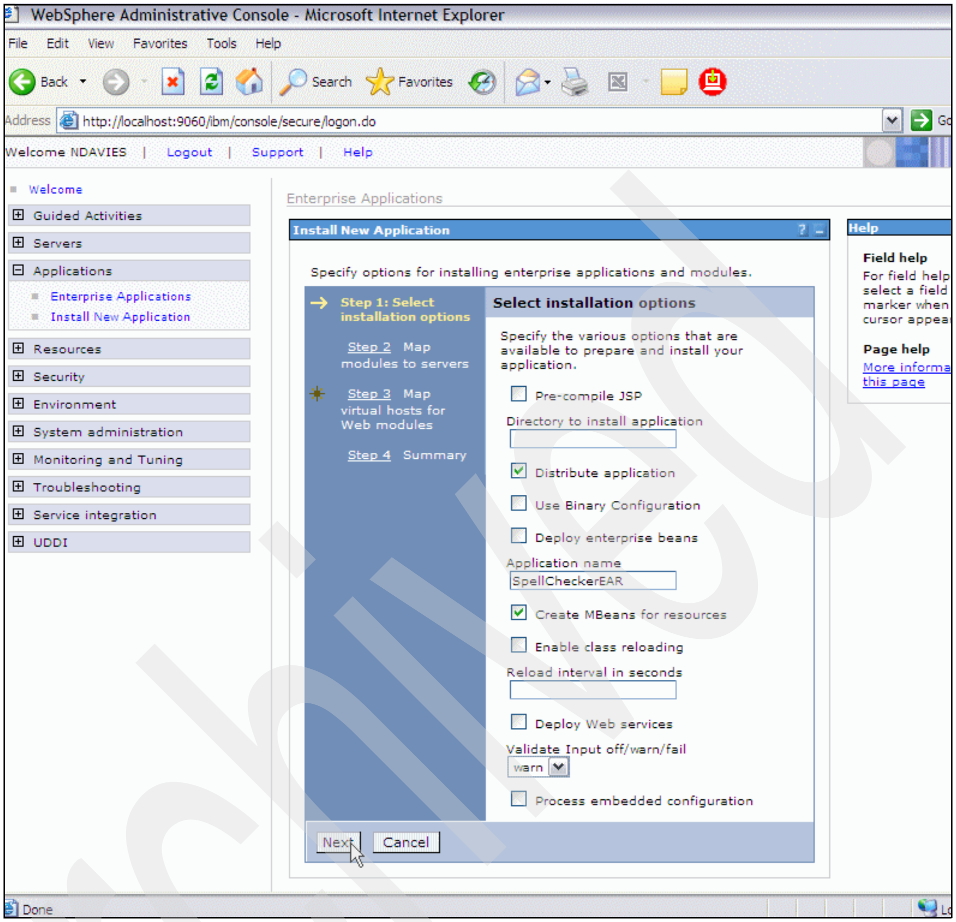
Figure 8-14 Installing SpellChecker.ear file - Step 2

Accept the defaults for existing bindings by clicking **Next**, as also shown previously in Figure 8-5 on page 315.

Accept any security warnings you receive by clicking **Continue** on the screen Application Security Warnings, as also shown previously in Figure 8-6 on page 316.

Accept the defaults on the screen Select installation options by clicking **Next**, as shown in Figure 8-15 on page 325.

Note: Do *not* change the application name SpellCheckerEAR.



Install New Application

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Map virtual hosts for Web modules

Step 4 Summary

Select installation options

Specify the various options that are available to prepare and install your application.

□ Pre-compile JSP

Directory to install application

☑ Distribute application

□ Use Binary Configuration

□ Deploy enterprise beans

Application name: SpellCheckerEAR

☑ Create MBeans for resources

□ Enable class reloading

Reload interval in seconds

□ Deploy Web services

Validate Input off/warn/fail: warn

□ Process embedded configuration

Next Cancel

Field help
For field help select a field marker when cursor appea

Page help
More informa this page

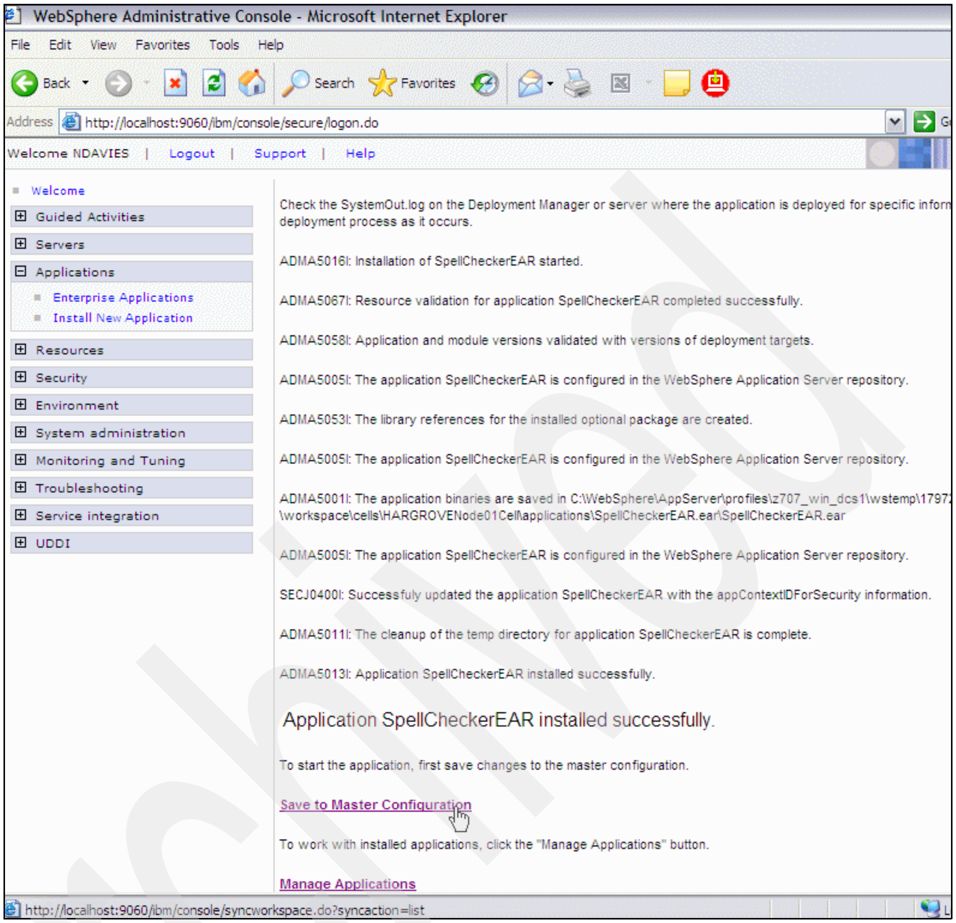
Figure 8-15 Installing SpellChecker.ear file - Step 3

Now accept the defaults on the screen Map modules to servers by clicking **Next**, as also shown previously in Figure 8-8 on page 318.

Accept the defaults on the screen Map virtual hosts for Web modules by clicking the **Next** button, as also shown previously in Figure 8-9 on page 319.

On the “Install new application summary” screen, review your installation choices and proceed by clicking **Next**, as shown in Figure 8-10 on page 320.

After the application is successfully deployed, you should see a screen similar to Figure 8-16 on page 326. Click **Save to Master Configuration**.



Check the SystemOut.log on the Deployment Manager or server where the application is deployed for specific inforn
deployment process as it occurs.

ADMA5016I: Installation of SpellCheckerEAR started.

ADMA5067I: Resource validation for application SpellCheckerEAR completed successfully.

ADMA5058I: Application and module versions validated with versions of deployment targets.

ADMA5005I: The application SpellCheckerEAR is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application SpellCheckerEAR is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in C:\WebSphere\AppServer\profiles\z707_win_dcs1\wstemp\1797
\workspace\cells\HARGROVENode01Cell\applications\SpellCheckerEAR.ear\SpellCheckerEAR.ear

ADMA5005I: The application SpellCheckerEAR is configured in the WebSphere Application Server repository.

SECJ0400I: Successfuly updated the application SpellCheckerEAR with the appContextIDForSecurity information.

ADMA5011I: The cleanup of the temp directory for application SpellCheckerEAR is complete.

ADMA5013I: Application SpellCheckerEAR installed successfully.

Application SpellCheckerEAR installed successfully.

To start the application, first save changes to the master configuration.

Save to Master Configuration

To work with installed applications, click the "Manage Applications" button.

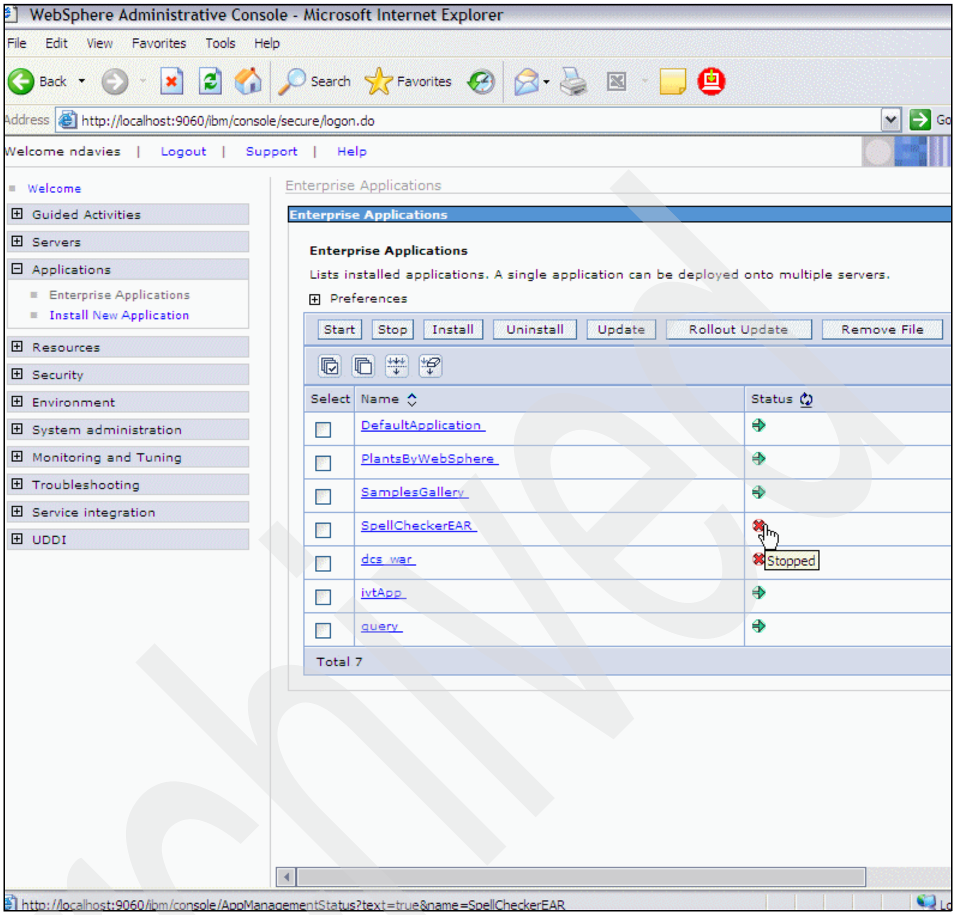
Manage Applications

http://localhost:9060/ibm/console/syncworkspace.do?syncaction=list

Figure 8-16 Installing SpellChecker.ear file - Successful Deployment

As also shown previously in Figure 8-12 on page 322, click **Save** to update the WebSphere Application Server master configuration.

Verify that the SpellCheckerEAR application is now showing in the list of deployed enterprise applications in the WebSphere Application Server Administrative Console and that its status is showing as `Stopped`, as shown in Figure 8-17 on page 327. Allow the application to remain stopped for the time being.



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
□	DefaultApplication	➔
□	PlantsByWebSphere	➔
□	SamplesGallery	➔
□	SpellCheckerEAR	✖
□	dcs_war	✖
□	ivtApp	➔
□	query	➔
Total 7		

Figure 8-17 Viewing the deployed SpellCheckEAR application

8.3.4 Configuring DCS

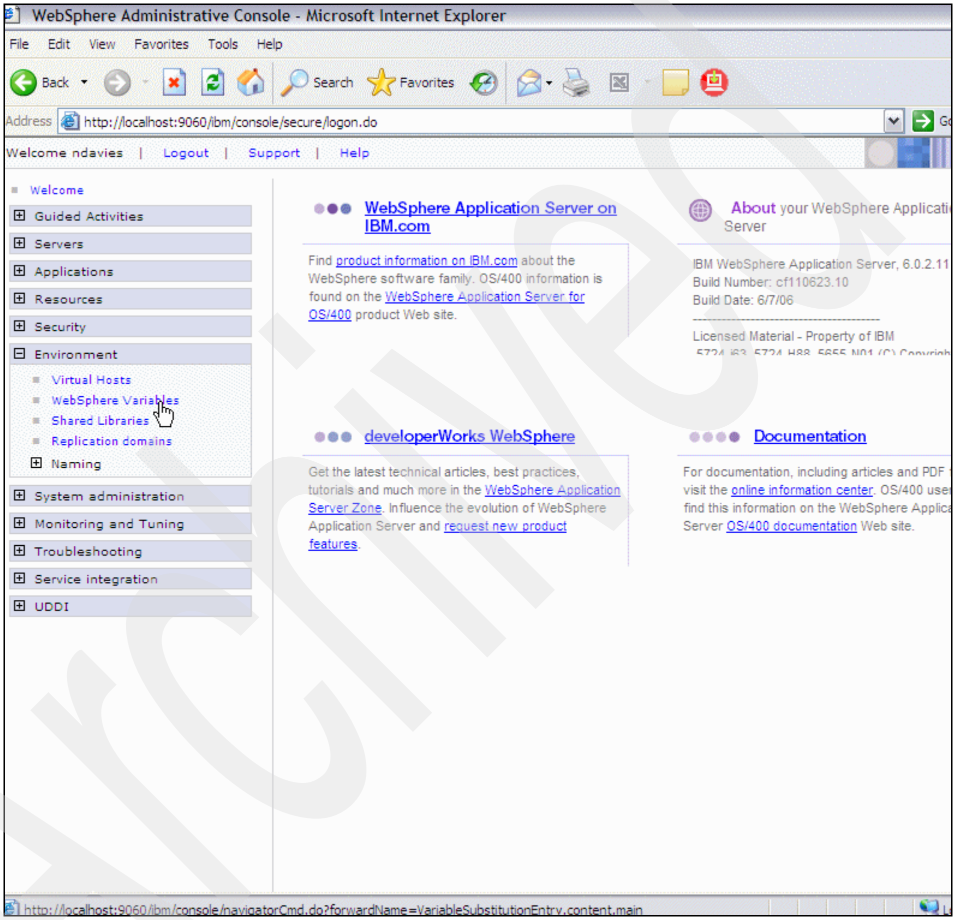
The remaining tasks to be completed on the remote DCS server are as follows:

- Setting WebSphere variables
- Executing the DCS configuration script

Setting WebSphere variables

In order to configure remote DCS services, you need to define two new WebSphere variables.

Using the WebSphere Application Server Administrative Console on the remote DCS server, click **Environment/WebSphere Variables**, as shown in Figure 8-18.



WebSphere Administrative Console - Microsoft Internet Explorer

http://localhost:9060/ibm/console/secure/logon.do

Welcome ndavies | Logout | Support | Help

WebSphere Application Server on IBM.com

Find product information on IBM.com about the WebSphere software family. OS/400 information is found on the WebSphere Application Server for OS/400 product Web site.

About your WebSphere Applicati Server

IBM WebSphere Application Server, 6.0.2.11
Build Number: cf110623.10
Build Date: 6/7/06

Licensed Material - Property of IBM

developerWorks WebSphere

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

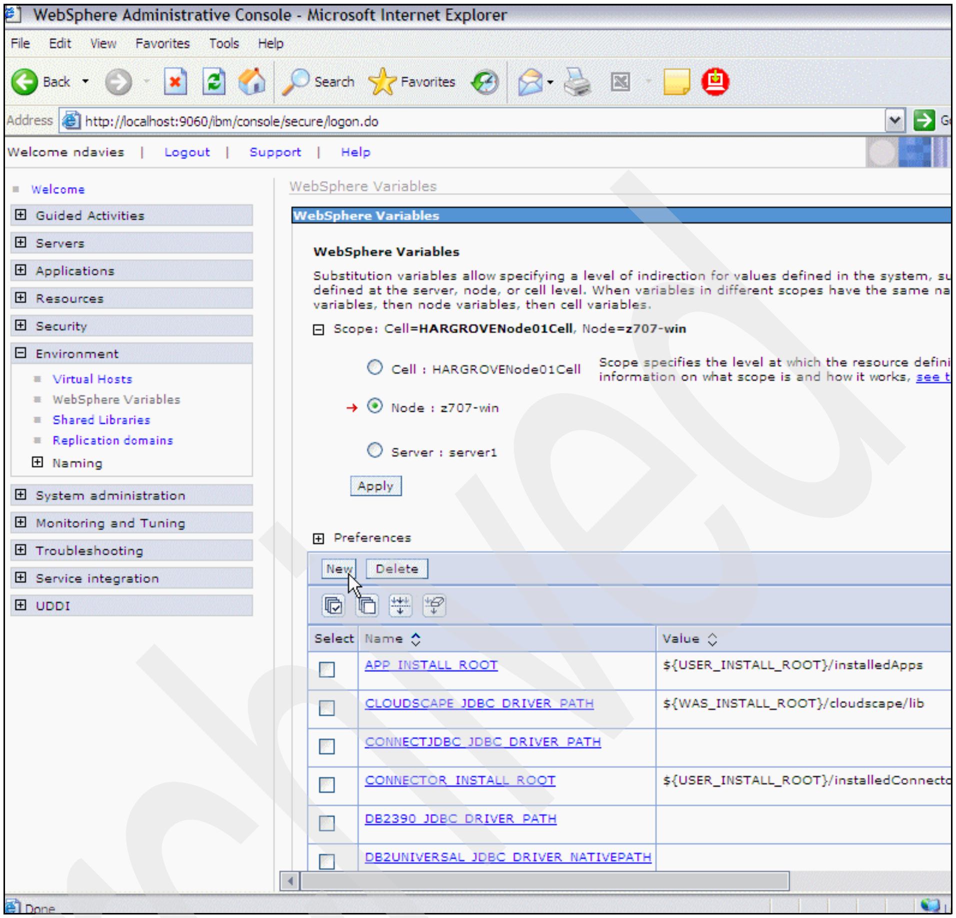
Documentation

For documentation, including articles and PDF visit the online information center. OS/400 use find this information on the WebSphere Applica Server OS/400 documentation Web site.

http://localhost:9060/ibm/console/navigatorCmd.do?forwardName=VariableSubstitutionEntry.content.main

Figure 8-18 Setting WebSphere variables for DCS - Step 1

Now set the scope to the node level (in our case, z707-win) and click **New** to create a new WebSphere variable, as shown in Figure 8-19 on page 329.



WebSphere Administrative Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:9060/ibm/console/secure/logon.do

Welcome ndavies | Logout | Support | Help

- Welcome
- ⊞ Guided Activities
- ⊞ Servers
- ⊞ Applications
- ⊞ Resources
- ⊞ Security
- ⊟ Environment
  - Virtual Hosts
  - WebSphere Variables
  - Shared Libraries
  - Replication domains
  - ⊞ Naming
- ⊞ System administration
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

WebSphere Variables

WebSphere Variables

WebSphere Variables

Substitution variables allow specifying a level of indirection for values defined in the system, su defined at the server, node, or cell level. When variables in different scopes have the same na variables, then node variables, then cell variables.

⊟ Scope: Cell=**HARGROVENode01Cell**, Node=**z707-win**

○ Cell : HARGROVENode01Cell

→ ● Node : z707-win

○ Server : server1

Scope specifies the level at which the resource defini information on what scope is and how it works, see t

Apply

⊞ Preferences

New Delete

Select	Name	Value
□	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps
□	CLOUDSCAPE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/cloudscape/lib
□	CONNECTJDBC_JDBC_DRIVER_PATH	
□	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedConnecto
□	DB2390_JDBC_DRIVER_PATH	
□	DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH	

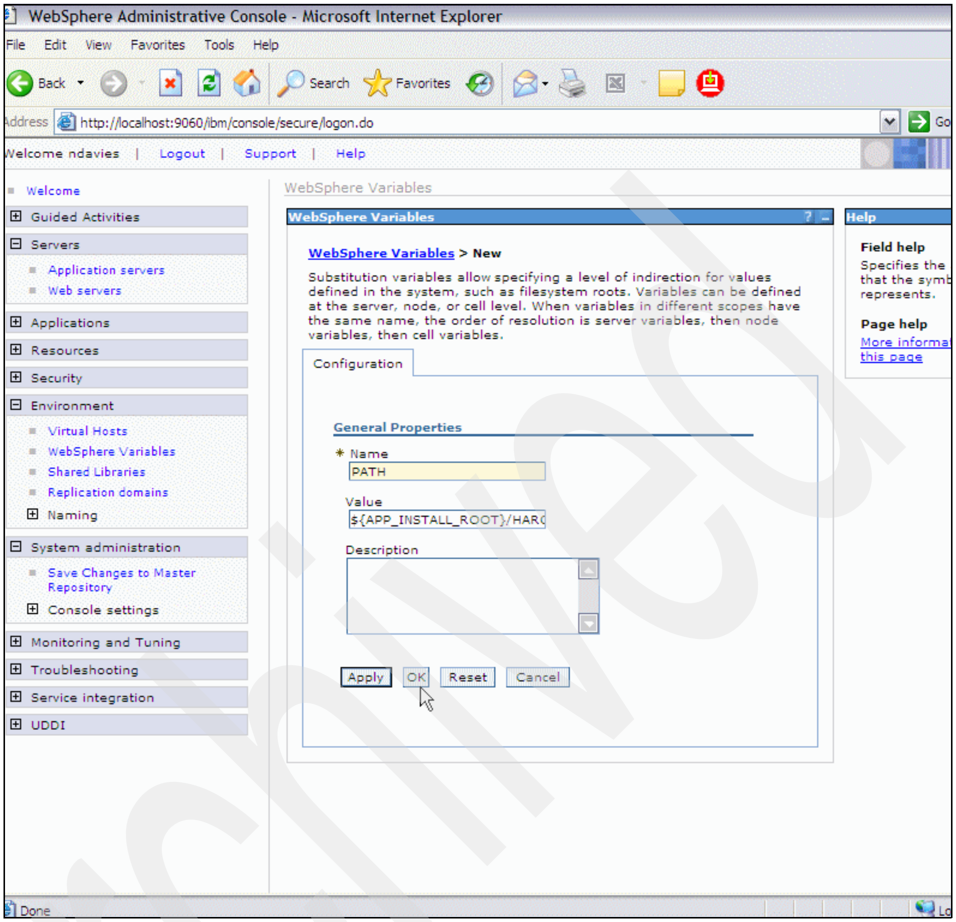
Done

Figure 8-19 Setting WebSphere variables for DCS - Step 2

Set the name of the new variable to `PATH` and the value of the variable to `${APP_INSTALL_ROOT}/`*`cellname`*`/dcs_war.ear/dcs.war/WEB-INF/lib/oiexport`, where *`cellname`* is the name of the WebSphere Application Server cell where you have deployed the dcs_war application on the remote DCS server.

In our case, after substituting the cell name, we used `${APP_INSTALL_ROOT}/HARGROVENode01Cell/dcs_war.ear/dcs.war/WEB-INF/lib/oiexport`.

After setting the name and the value of the new variable, click **OK** to save the new variable, as shown in Figure 8-20 on page 330.



WebSphere Variables > New

Substitution variables allow specifying a level of indirection for values defined in the system, such as filesystem roots. Variables can be defined at the server, node, or cell level. When variables in different scopes have the same name, the order of resolution is server variables, then node variables, then cell variables.

Configuration

General Properties

* Name: PATH

Value: ${APP_INSTALL_ROOT}/HAR(

Description:

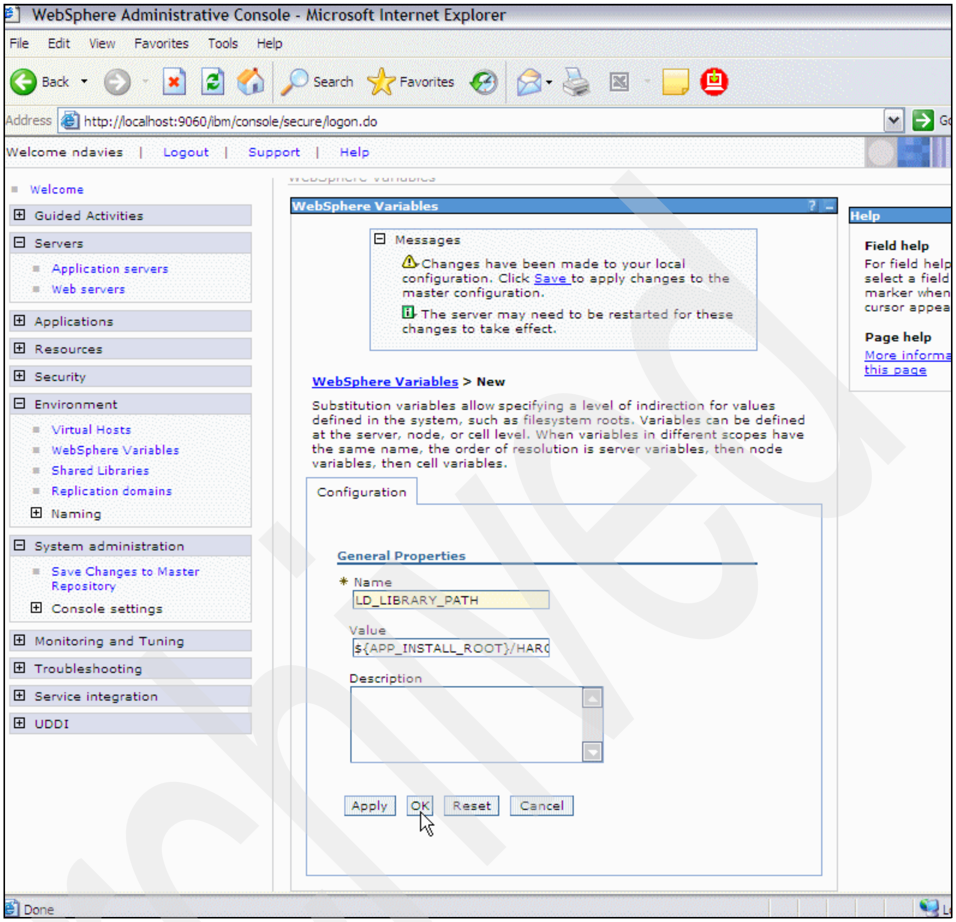
Apply OK Reset Cancel

Figure 8-20 Setting WebSphere environment variables for DCS - Step 3

Note: The environment variable we are defining includes a recursive reference to another WebSphere variable: ${APP_INSTALL_ROOT}.

In fact, the value of this second variable is also visible in Figure 8-19 on page 329.

Now repeat the process to define a second new variable with the name `LD_LIBRARY_PATH` and the same value as before `${APP_INSTALL_ROOT}/`*`cellname`*`/dcs_war.ear/dcs.war/WEB-INF/lib/oiexport`, where *`cellname`* is the name of the WebSphere Application Server cell where you have deployed the dcs_war application on the remote DCS server. Click **OK** to save the second variable, as shown in Figure 8-21 on page 331.



Messages

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

ℹ The server may need to be restarted for these changes to take effect.

WebSphere Variables > New

Substitution variables allow specifying a level of indirection for values defined in the system, such as filesystem roots. Variables can be defined at the server, node, or cell level. When variables in different scopes have the same name, the order of resolution is server variables, then node variables, then cell variables.

Configuration

General Properties

* Name: LD_LIBRARY_PATH

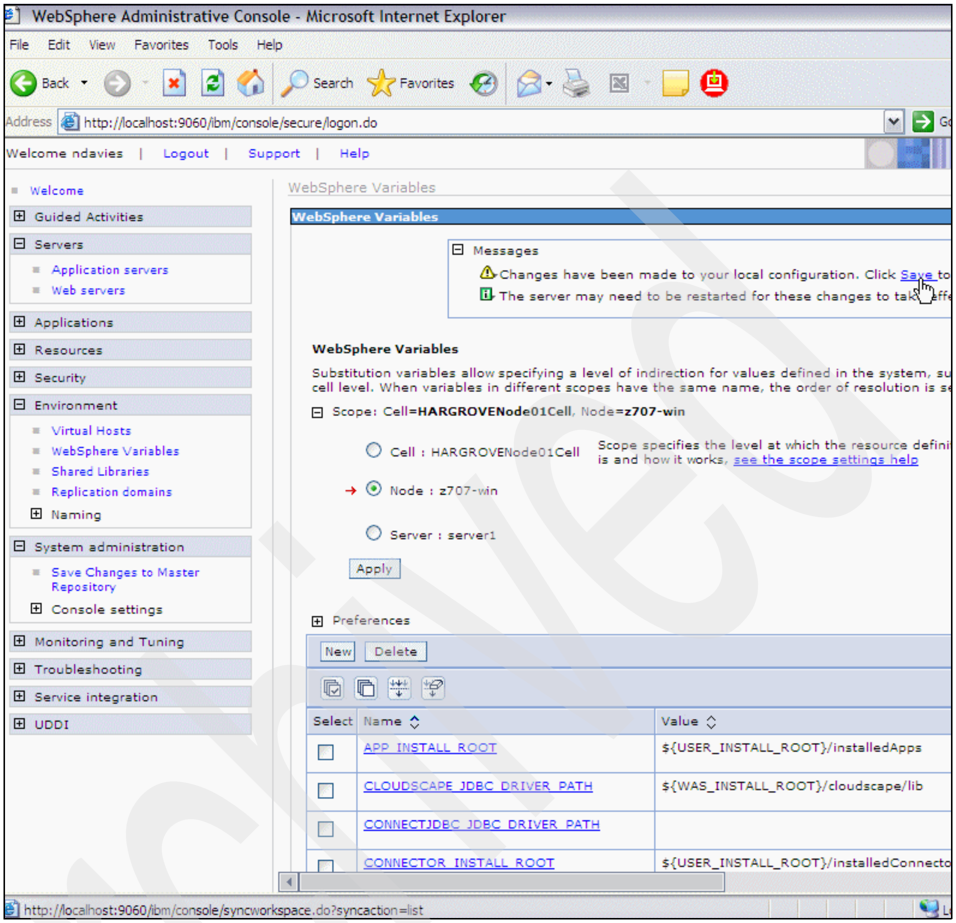
Value: ${APP_INSTALL_ROOT}/HARC

Description:

Apply OK Reset Cancel

Figure 8-21 Setting WebSphere variables for DCS - Step 4

Having created both new variables, save the changes to the master configuration, as shown in Figure 8-22 on page 332.



WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. Click Save to

ℹ The server may need to be restarted for these changes to take effe

WebSphere Variables

Substitution variables allow specifying a level of indirection for values defined in the system, su cell level. When variables in different scopes have the same name, the order of resolution is s

Scope: Cell=**HARGROVENode01Cell**, Node=**z707-win**

○ Cell : HARGROVENode01Cell

Scope specifies the level at which the resource defini is and how it works, see the scope settings help

→ ● Node : z707-win

○ Server : server1

Apply

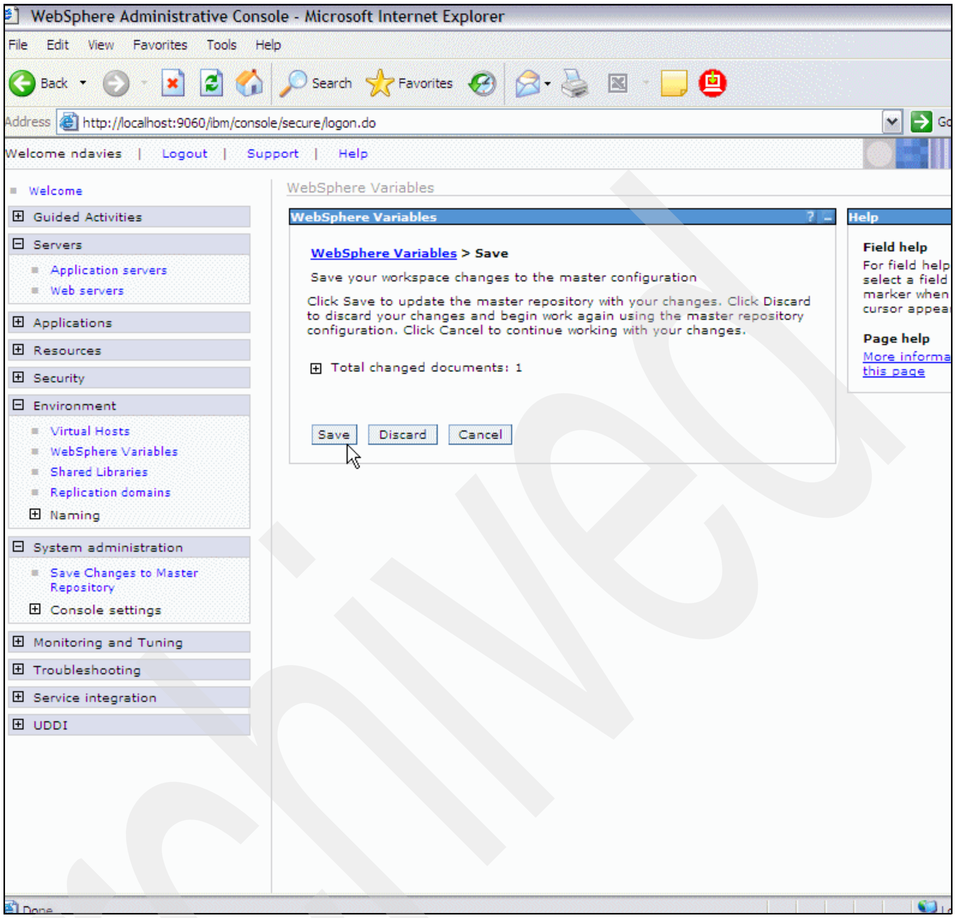
⊞ Preferences

New Delete

Select	Name ⇕	Value ⇕
□	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps
□	CLOUDSCAPE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/cloudscape/lib
□	CONNECTJDBC_JDBC_DRIVER_PATH	
□	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedConnecto

Figure 8-22 Setting WebSphere variables - Saving the master configuration

Finally, save the changes to the master configuration as shown in Figure 8-23 on page 333.



WebSphere Variables > Save

Save your workspace changes to the master configuration

Click Save to update the master repository with your changes. Click Discard to discard your changes and begin work again using the master repository configuration. Click Cancel to continue working with your changes.

⊞ Total changed documents: 1

Figure 8-23 WebSphere variables - Saving the master configuration

Executing the DCS configuration script

On the remote DCS server, use the **cd** command to navigate to the temporary directory where you unzipped the files from the remoteDCS.zip file in 8.3.1, “Locate remote DCS and Spell Checker components” on page 310. Run the script setupdcs.bat for Windows platforms (or setupremotedcs.sh for all other platforms) as shown in Example 8-2 on page 334. You are prompted to supply certain configuration parameters as shown in the example.

Note: You are prompted for the WebSphere Application Server profile directory where the DCS.war application is installed, but the prompt is misleading.

Do *not* reply with the directory name for that WebSphere Application Server profile. Instead, reply with the WebSphere Application Server profile name itself. In our case, this is z707_win_dcs1. The script will construct the directory name correctly itself using the other configuration parameters.

Example 8-2 Executing the DCS configuration script

```
Z:\>cd \dcs
Z:\dcs>setupdcs
Enter the directory where the IBM WebSphere Application Server is
installed:
C:\WebSphere\AppServer
Enter the profile directory where DCS.war is installed:
z707_win_dcs1
Enter the IBM WebSphere Application Server cellname where DCS is
installed:
HARGROVENode01Cell
Please wait, set up is now copying Stellant files to
"C:\WebSphere\AppServer\profiles\z707_win_dcs1\installedApps\HARGROVENo
de01Cell\dcs_war.ear\dcs.war\WEB-INF\lib"
232 File(s) copied

   Setup complete.  Please restart WebSphere Application Server
   in order to use Document Conversion Services.

Z:\dcs>
```

Now verify that the oiexport directory was added to the directory identified by the setupdcs configuration script for the target of the Stellant files. In our case, this is C:\WebSphere\AppServer\profiles\z707_win_dcs1\installedApps\HARGROVENode01Cell\dcs_war.ear\dcs.war\WEB-INF\lib.

We show the contents of our directory after running the setupdcs configuration script in Example 8-3 on page 335.

Example 8-3 Confirming the oiexport directory

```
Directory of
C:\WebSphere\AppServer\profiles\z707_win_dcs1\installedApps\HARGRO
VENode01Cell\dcs_war.ear\dcs.war\WEB-INF\lib

03/05/2007  06:50 PM    <DIR>          .
03/05/2007  06:50 PM    <DIR>          ..
12/01/2006  03:08 PM            62,694
avalon-framework-cvs-20020806.jar
12/01/2006  03:08 PM         2,108,563 batik.jar
12/01/2006  03:08 PM         1,975,923 convertors.jar
12/01/2006  03:08 PM            11,658 convertors.xml
12/01/2006  03:08 PM            19,765 export.cfg
12/01/2006  03:08 PM            23,570 Export.jar
12/01/2006  03:08 PM         1,519,742 fop.jar
10/19/2006  10:18 PM           301,761 html-parser-1.4.4.jar
12/01/2006  03:08 PM           810,035 jakarta-poi.jar
12/01/2006  03:08 PM             5,203 no_stellent.xml
03/05/2007  06:50 PM    <DIR>          oiexport
              10 File(s)      6,838,914 bytes
               3 Dir(s)  55,394,525,184 bytes free
```

Note: On Linux or z/Linux servers, you will not be able to launch setupremotedcs.sh as supplied because it contains Windows CR/LF characters at the end of the lines. When you launch the script, it fails with the following message:

```
: bad interpreter: No such file or directory
```

The solutions to this problem are:

Solution 1: If you are using something like UltraVNC that supports X11 graphics, then open the setupremotedcs.sh in the kwrite editor. In kwrite, go to **Tools** -> **End of Line**. You will see that Windows-Dos is checked. Check the UNIX box instead and then save the file. Checking the UNIX box will change the CR/LF to UNIX CR. Now you can launch the modified setupremotedcs.sh script.

Solution 2: On many UNIX systems, including z/Linux, there is are dos2unix and unix2dos utilities that can change the file format to and from UNIX file format. On HP-UX, there are dos2ux and ux2dos utilities.

The following command sequence (input shown in bold) shows how to change the setupremotedcs.sh to UNIX format using the dos2unix utility.

```
machine2:/opt/IBM/Portaldcs/dcs # cp setupremotedcs.sh
setupremotedcs.orig.sh

machine2:/opt/IBM/Portaldcs/dcs # ./setupremotedcs.sh: bad
interpreter: No such file or directory

machine2:/opt/IBM/Portaldcs/dcs # dos2unix setupremotedcs.sh

dos2unix: converting file test.sh to UNIX format ...

machine2:/opt/IBM/Portaldcs/dcs # ./setupremotedcs.sh

bash: ./setupremotedcs.sh: Permission denied

machine2:/opt/IBM/Portaldcs/dcs # chmod 775 setupremotedcs.sh

machine2:/opt/IBM/Portaldcs/dcs # ./setupremotedcs.sh....

Now the setupdcs.sh script will run successfully
```

8.3.5 Restart WebSphere Application Server on the remote DCS server

After completing all the other tasks described in 8.3, “Configuring DCS and Spell Checker on the remote server” on page 310, stop the WebSphere Application Server task on the remote DCS server. After the server task has stopped, then restart the server.

8.4 WCM configuration on the Portal on z/OS

The configuration of the WCM components on the Portal for z/OS consists of the following steps, which must be performed in the sequence shown on the z/OS system where Portal is installed.

- Configuring the content properties file
- Configuring the Portal server to use remote DCS services
- Deploying the authoring portlets

8.4.1 Configuring the content properties file

In order to ensure that document conversions will work correctly on the remote DCS server, you must configure the content-types.properties file on the z/OS system where WebSphere Portal is installed.

We illustrate the process in this section, but basic instructions can also be found in the WebSphere Portal for z/OS Information Center (IC) at:

`http://www.ibm.com/WebSphere/portal/library`

Navigate to the Information Center section **Managing content** -> **Managing documents** -> **Preparing to work with documents** -> **Enabling Document Conversion Services**.

> **Tip:** The direct link to this section is currently available at the following site:
>
> `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/dcs_info.html`
>
> This link could change over time, but you can always navigate to the correct section by using the section titles described above.

You can choose to support the conversion of some or all documents of the following types:

- Microsoft Office
- Lotus SmartSuite®
- OpenOffice

When you have decided which document type conversions you wish to support, copy from the Information Center the required entries for each document type.

To begin, you need to locate the content-types.properties file in the directory <AppserverRoot>/java/lib on the z/OS system where WebSphere Portal is installed and copy it to the directory <AppserverRoot>/properties with a new file name of content-types.properties.EBCDIC.

In our case, we used an OMVS shell to copy from /waswpconfig/wpcell/wpnode/AppServer/java/lib to the directory /waswpconfig/wpcell/wpnode/AppServer/properties, as shown in Example 8-4.

Example 8-4 Copying the content-types.properties file

```
DAVIES @ SC49:/u/davies>cd
/waswpconfig/wpcell/wpnode/AppServer/java/lib
DAVIES @ SC49:/SC49/waswpconfig/wpcell/wpnode/AppServer/java/lib>cp
content-types.properties
../../properties/content-types.properties.EBCDIC
DAVIES @ SC49:/SC49/waswpconfig/wpcell/wpnode/AppServer/java/lib>
```

Next, you need to copy the information from the Information Center for the document type conversions you wish to support and append this to the file content-types.properties.EBCDIC. Choose a way to append this information that you are familiar with.

You can use the ISPF editor to update the file. Alternatively, you can FTP the file to your workstation and do the edits there and then FTP the file back again after editing. If you decide on the FTP approach, ensure that you use an ASCII FTP in both directions.

We show an excerpt of the content-types.properties.EBCDIC file before editing in Example 8-5.

Example 8-5 Excerpt from the file content-types.properties.EBCDIC

```
#
# @(#)src/net/pfm/content-types.properties, net, am142, 20060824 1.5.2.1
# ===========================================================================
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
#
# IBM SDK, Java(tm) 2 Technology Edition, v1.4.2
# (C) Copyright IBM Corp. 1998, 2004. All Rights Reserved
# ============================================================================
#

#sun.net.www MIME content-types table; version 1.6, 05/04/99
#
# Property fields:
#
#   <description> ::= 'description' '=' <descriptive string>
#    <extensions> ::= 'file_extensions' '=' <comma-delimited list, include '.'>
#         <image> ::= 'icon' '=' <filename of icon image>
#        <action> ::= 'browser' | 'application' | 'save' | 'unknown'
#   <application> ::= 'application' '=' <command line template>
#

#
# The "we don't know anything about this data" type(s).
# Used internally to mark unrecognized types.
#
content/unknown: description=Unknown Content
unknown/unknown: description=Unknown Data Type

#
# The template we should use for temporary files when launching an application
# to view a document of given type.
#
temp.file.template: /tmp/%s

#
# The "real" types.
#
application/octet-stream: \
   description=Generic Binary Stream;\
   file_extensions=.saveme,.dump,.hqx,.arc,.o,.a,.bin,.exe,.z,.gz

application/oda: \
   description=ODA Document;\
   file_extensions=.oda

application/pdf: \
   description=Adobe PDF Format;\
   file_extensions=.pdf

application/postscript: \
   description=Postscript File;\
   file_extensions=.eps,.ai,.ps;\
   icon=ps;\
   action=application;\
   application=imagetool %s
```

```
...............................

video/x-sgi-movie: \
   description=SGI Movie;\
   file_extensions=.movie,.mv

message/rfc822: \
   description=Internet Email Message;\
   file_extensions=.mime

application/xml: \
   description=XML document;\
   file_extensions=.xml
```

When editing this file, we append all the property lines copied from the Information Center to the file content-types.properties.EBCDIC to support all available document types. The appended lines are shown in Example 8-6.

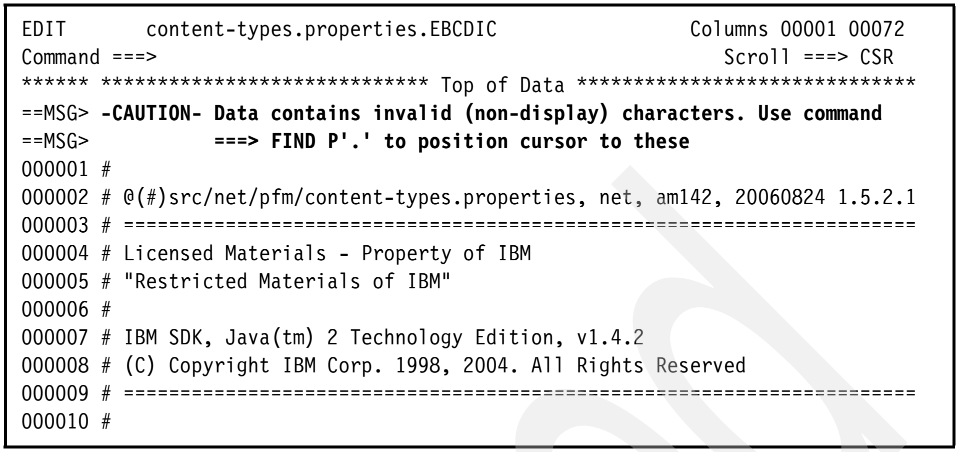
> **Note:** The edits are case-sensitive.

Example 8-6 Adding properties for remote document conversion

```
#
# Additional types added for dcs N.Davies 03/09/07
#
application/msword: \
 description=Microsoft Word;\
 file_extensions=.doc
application/vnd.ms-excel: \
 description=Microsoft Excel;\
 file_extensions=.xls
application/vnd.ms-powerpoint: \
 description=Microsoft PowerPoint;\
 file_extensions=.ppt
application/vnd.lotus-freelance: \
 description=Lotus Freelance;\
 file_extensions=.prz
application/vnd.lotus-1-2-3: \
 description=Lotus 1-2-3;\
 file_extensions=.123
application/vnd.lotus-wordpro: \
 description=Lotus WordPro;\
 file_extensions=.lwp
application/vnd.sun.xml.writer: \
 description=Open Office;\
 file_extensions=.sxw
```

```
application/vnd.sun.xml.writer.template: \
 description=Open Office;\
 file_extensions=.stw
application/vnd.sun.xml.writer.global: \
 description=Open Office;\
 file_extensions=.sxg
application/vnd.sun.xml.calc: \
 description=Open Office;\
 file_extensions=.sxc
application/vnd.sun.xml.calc.template: \
 description=Open Office;\
 file_extensions=.stc
application/vnd.sun.xml.impress: \
 description=Open Office;\
 file_extensions=.sxi
application/vnd.sun.xml.impress.template: \
 description=Open Office;\
 file_extensions=.sti
application/vnd.sun.xml.draw: \
 description=Open Office;\
 file_extensions=.sxd
application/vnd.sun.xml.draw.template: \
 description=Open Office;\
 file_extensions=.std
application/vnd.sun.xml.math: \
 description=Open Office;\
 file_extensions=.sxm
```

If you use the ISPF editor to edit this file, you may encounter a warning that the file contains invalid (non-display) characters, as shown in Figure 8-24 on page 342.



```
EDIT       content-types.properties.EBCDIC              Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data *****************************
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG>           ===> FIND P'.' to position cursor to these
000001 #
000002 # @(#)src/net/pfm/content-types.properties, net, am142, 20060824 1.5.2.1
000003 # =======================================================================
000004 # Licensed Materials - Property of IBM
000005 # "Restricted Materials of IBM"
000006 #
000007 # IBM SDK, Java(tm) 2 Technology Edition, v1.4.2
000008 # (C) Copyright IBM Corp. 1998, 2004. All Rights Reserved
000009 # =======================================================================
000010 #
```

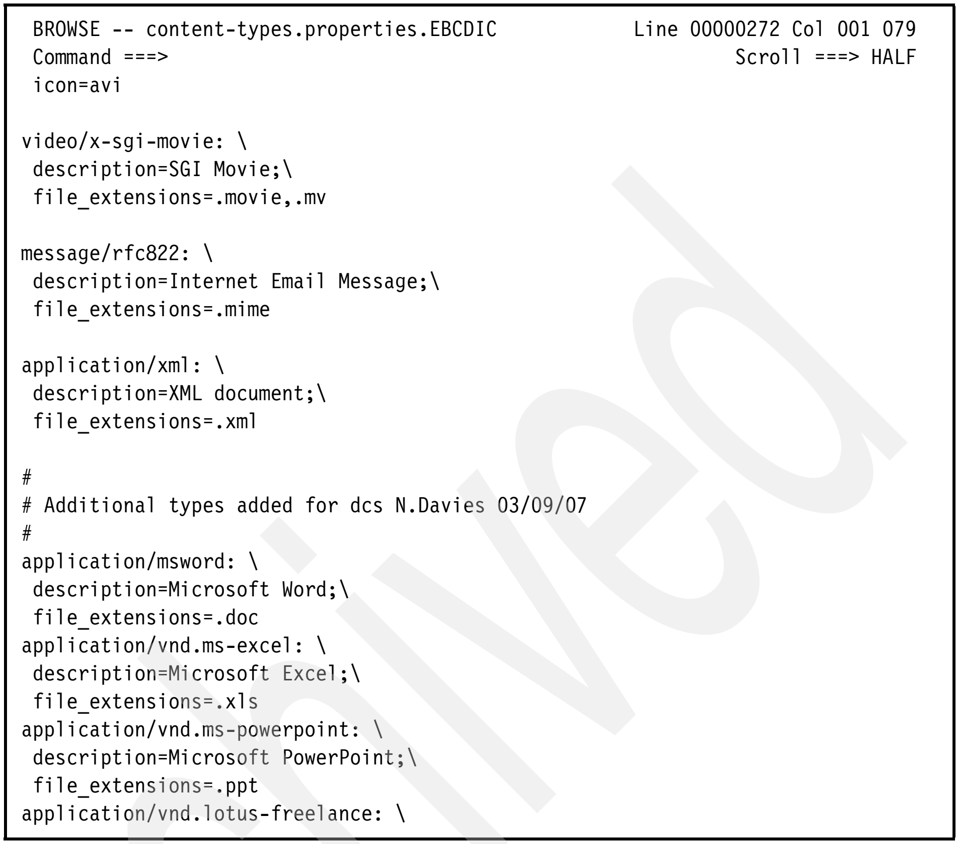
Figure 8-24 Editing the content-types.properties.EBCDIC file with ISPF

These “invalid” characters are X’05’ and probably remnants from an occasion when the file was edited using a different editor. You can remove them easily using the ISPF edit primary command **`C ALL Xc all x‘05’ x‘40’ 1`** which will change all x’05’ characters to blanks (X’40) if they occur in column 1.

If you then issue the ISPF edit primary command **F ALL P’.’** to find all non-display characters, the number found should now be zero. (The character in quotes in the previous command is a period.)

> **Note:** However you choose to edit this file, it is critical that you do *not* mix ASCII and EBCDIC values in the content-types.properties.EBCDIC.

If you edited the file on your local workstation and transferred it back to z/OS, you can confirm that your edits were correct by using the ISPF browse command as shown in Figure 8-25 on page 343, which shows a correctly edited file. If you have mixed code pages (ASCII and EBCDIC), not all characters will be readable.



```
 BROWSE -- content-types.properties.EBCDIC           Line 00000272 Col 001 079
 Command ===>                                                 Scroll ===> HALF
 icon=avi

video/x-sgi-movie: \
 description=SGI Movie;\
 file_extensions=.movie,.mv

message/rfc822: \
 description=Internet Email Message;\
 file_extensions=.mime

application/xml: \
 description=XML document;\
 file_extensions=.xml

#
# Additional types added for dcs N.Davies 03/09/07
#
application/msword: \
 description=Microsoft Word;\
 file_extensions=.doc
application/vnd.ms-excel: \
 description=Microsoft Excel;\
 file_extensions=.xls
application/vnd.ms-powerpoint: \
 description=Microsoft PowerPoint;\
 file_extensions=.ppt
application/vnd.lotus-freelance: \
```

Figure 8-25 Browsing a correctly edited content-types.properties.EBCDIC file

You need to use the **iconv** command to convert the content-types.properties.EBCDIC file to ASCII for use by WebSphere Application Server. You issue this command from an OMVS shell, as shown in Example 8-7. The new file in ASCII format should be called content-types.properties.

Example 8-7 Converting the content-types.properties.EBCDIC to ASCII

```
DAVIES @ SC49:/u/davies>cd
/SC49/waswpconfig/wpcell/wpnode/AppServer/properties
DAVIES @
SC49:/SC49/waswpconfig/wpcell/wpnode/AppServer/properties>iconv -f
IBM-1047 -t ISO8859-1 content-types.properties.EBCDIC >
content-types.properties
```

A correctly converted file will be unreadable with ISPF browse, but will be readable if you download the file to your local workstation using a binary FTP.

Finally, you need to set the security attributes on the file you created using the **chmod** and **chown** commands as shown in Example 8-8, where WPADMIN is your WebSphere Application Server administrative user ID and WPCFG is your WebSphere Application Server configuration group ID.

Note: To issue the **chown** command shown in Example 8-8, you need to either be logged on as the user WPADMIN and be a member of the group WPCFG, or have root user authority in OMVS.

Example 8-8 Setting security attributes on the content-types.properties file

```
DAVIES @ SC49:/SC49/waswpconfig/wpcell/wpnode/AppServer/properties>
chmod 775 content-types.properties
DAVIES @ SC49:/SC49/waswpconfig/wpcell/wpnode/AppServer/properties>
chown WPADMIN:WPCFG content-types.properties
```

Important: You must repeat the steps provided in 8.4.1, “Configuring the content properties file” on page 337 for *each* managed node in a Network Deployment cell. To save time you can copy the edited content-types.properties file from one node <AppserverRoot>/properties directory to the equivalent directory for the other nodes, and edit the content-types.properties file only once.

Tip: If you use OMVS to do the copy, you can use the **-p** option on the copy command **cp** to also copy the permissions.

8.4.2 Configuring the Portal server to use remote DCS services

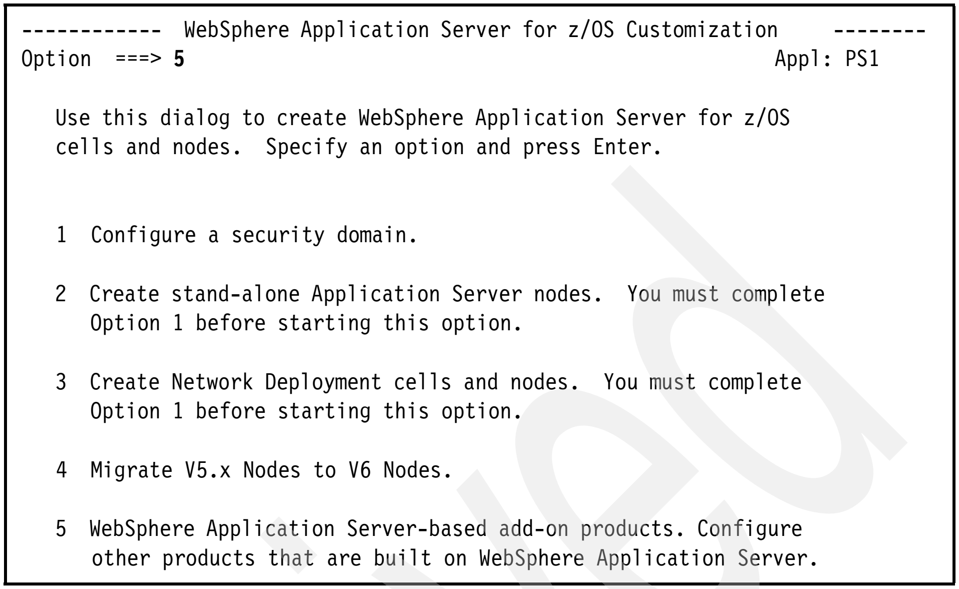
To configure WebSphere Portal on z/OS to use remote DCS services, you use the same ISPF customization dialogs which you used in 2.3, “Running the WebSphere Portal install dialog” on page 41.

Go to ISPF option 6 and enter the command as shown in Example 8-9 (replacing BBWP6049 with your high level qualifier) to start the install dialog.

Example 8-9 Start the WebSphere Application Server customization ISPF dialog

```
exec 'BBWP6049.SBBOCLIB(BBOWSTRT)' 'APPL(PS1) PROD(EJP) PRODHLQ(BBWP6049)'
```

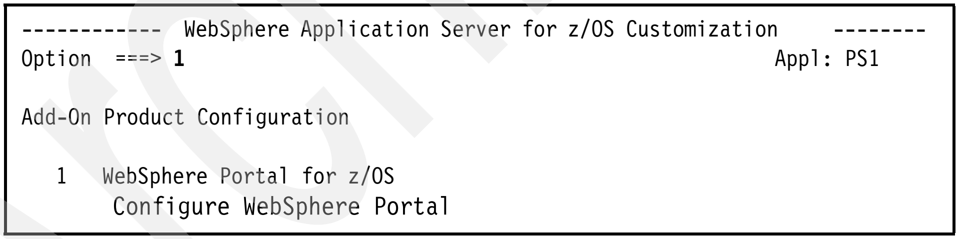
Choose option **5 - WebSphere Application Server-based add-on products** to configure products that are built onto WebSphere Application Server, as shown in Figure 8-26 on page 345.



```
------------  WebSphere Application Server for z/OS Customization     --------
Option  ===> 5                                                    Appl: PS1

   Use this dialog to create WebSphere Application Server for z/OS
   cells and nodes.  Specify an option and press Enter.


   1  Configure a security domain.

   2  Create stand-alone Application Server nodes.  You must complete
      Option 1 before starting this option.

   3  Create Network Deployment cells and nodes.  You must complete
      Option 1 before starting this option.

   4  Migrate V5.x Nodes to V6 Nodes.

   5  WebSphere Application Server-based add-on products. Configure
      other products that are built on WebSphere Application Server.
```

Figure 8-26 WebSphere Application Server customization primary navigation panel

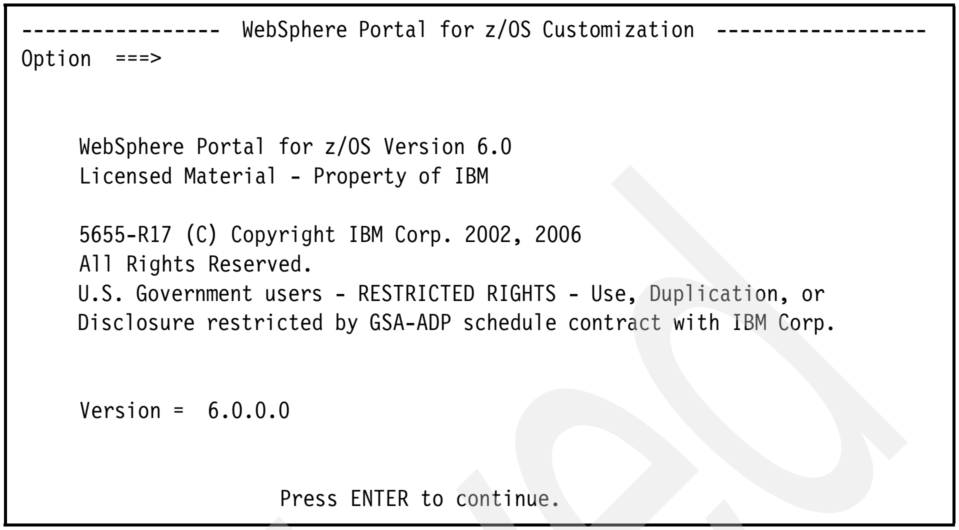
Choose **1 - WebSphere Portal for z/OS** from the list of WebSphere add-on products, as shown in Figure 8-27. We only had one choice in our scenario, but you may have more.



```
------------  WebSphere Application Server for z/OS Customization     --------
Option  ===> 1                                                    Appl: PS1

Add-On Product Configuration

   1   WebSphere Portal for z/OS
        Configure WebSphere Portal
```

Figure 8-27 Choose your WebSphere add-on product

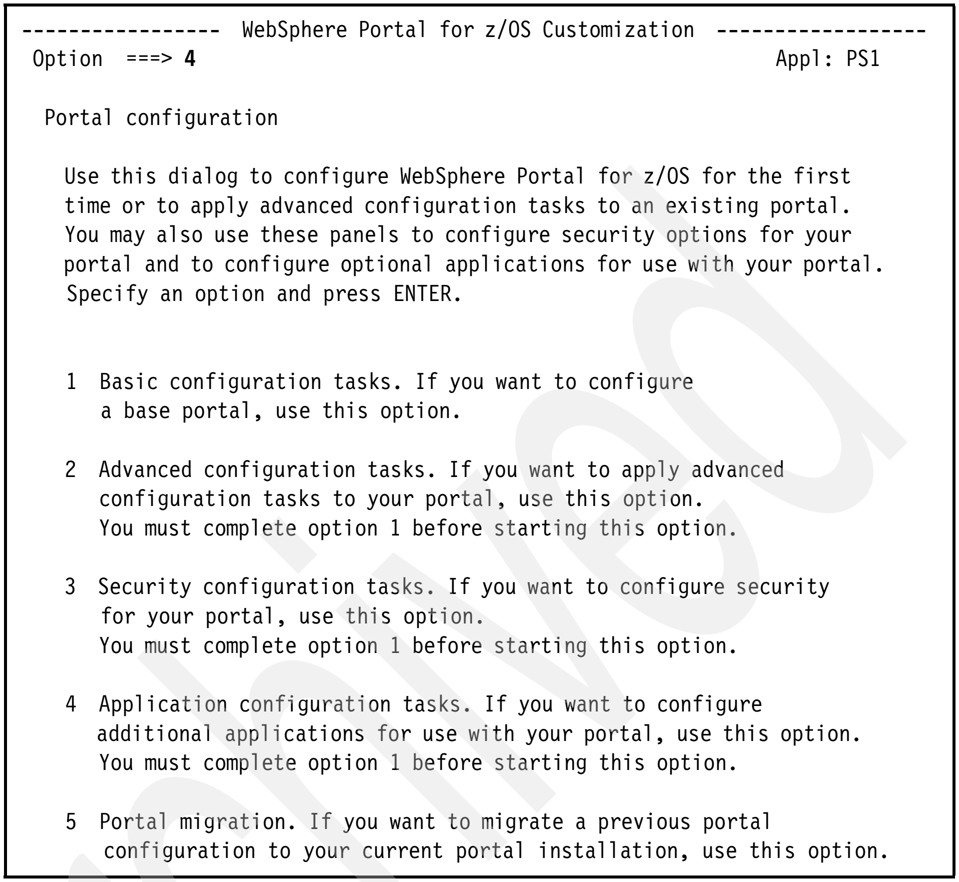
You may be presented with the license panel shown in Figure 8-28 on page 346. Verify the version before proceeding. Press Enter to accept the license agreement.



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



     WebSphere Portal for z/OS Version 6.0
     Licensed Material - Property of IBM

     5655-R17 (C) Copyright IBM Corp. 2002, 2006
     All Rights Reserved.
     U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
     Disclosure restricted by GSA-ADP schedule contract with IBM Corp.


     Version =  6.0.0.0


                        Press ENTER to continue.
```

Figure 8-28   License panel for Portal

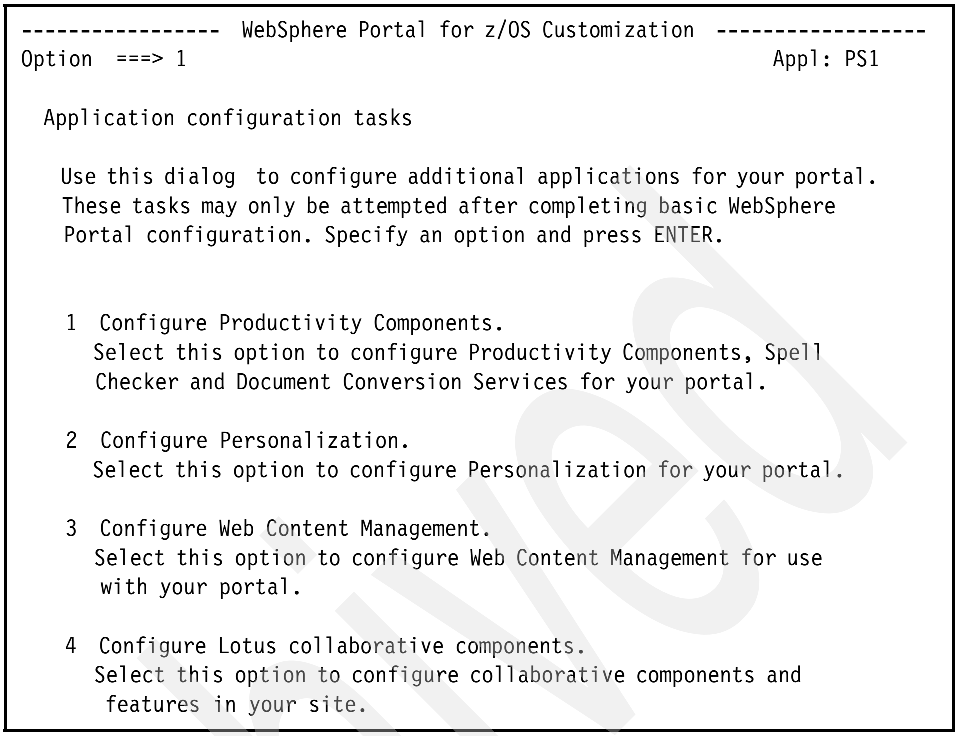
The screen Portal configuration shown in Figure 8-29 on page 347 is the main screen you use for all WebSphere Portal configuration tasks. Choose option **4 Application configuration tasks**.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> 4                                                   Appl: PS1

  Portal configuration

    Use this dialog to configure WebSphere Portal for z/OS for the first
    time or to apply advanced configuration tasks to an existing portal.
    You may also use these panels to configure security options for your
    portal and to configure optional applications for use with your portal.
    Specify an option and press ENTER.


    1  Basic configuration tasks. If you want to configure
       a base portal, use this option.

    2  Advanced configuration tasks. If you want to apply advanced
       configuration tasks to your portal, use this option.
       You must complete option 1 before starting this option.

    3  Security configuration tasks. If you want to configure security
       for your portal, use this option.
       You must complete option 1 before starting this option.

    4  Application configuration tasks. If you want to configure
       additional applications for use with your portal, use this option.
       You must complete option 1 before starting this option.

    5  Portal migration. If you want to migrate a previous portal
       configuration to your current portal installation, use this option.
```

Figure 8-29 Main Portal configuration

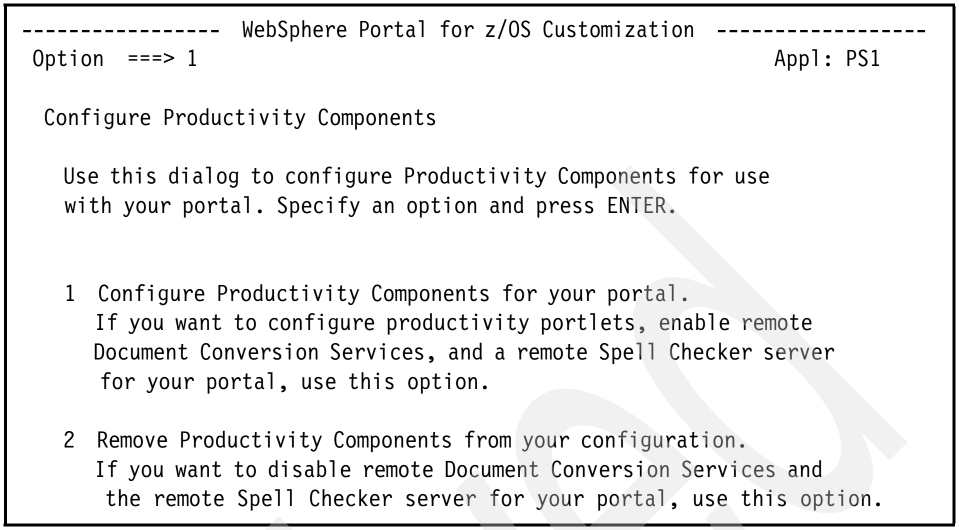
On the Portal screen Application configuration tasks shown in Figure 8-30 on page 348, select option **1 Configure Productivity Components**.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 1                                                  Appl: PS1

  Application configuration tasks

   Use this dialog  to configure additional applications for your portal.
   These tasks may only be attempted after completing basic WebSphere
   Portal configuration. Specify an option and press ENTER.


   1  Configure Productivity Components.
      Select this option to configure Productivity Components, Spell
      Checker and Document Conversion Services for your portal.

   2  Configure Personalization.
      Select this option to configure Personalization for your portal.

   3  Configure Web Content Management.
      Select this option to configure Web Content Management for use
       with your portal.

   4  Configure Lotus collaborative components.
      Select this option to configure collaborative components and
       features in your site.
```

Figure 8-30   Portal application configuration tasks

On the screen Configure productivity components shown in Figure 8-31 on page 349, select option **1 Configure Productivity Components for your portal**.

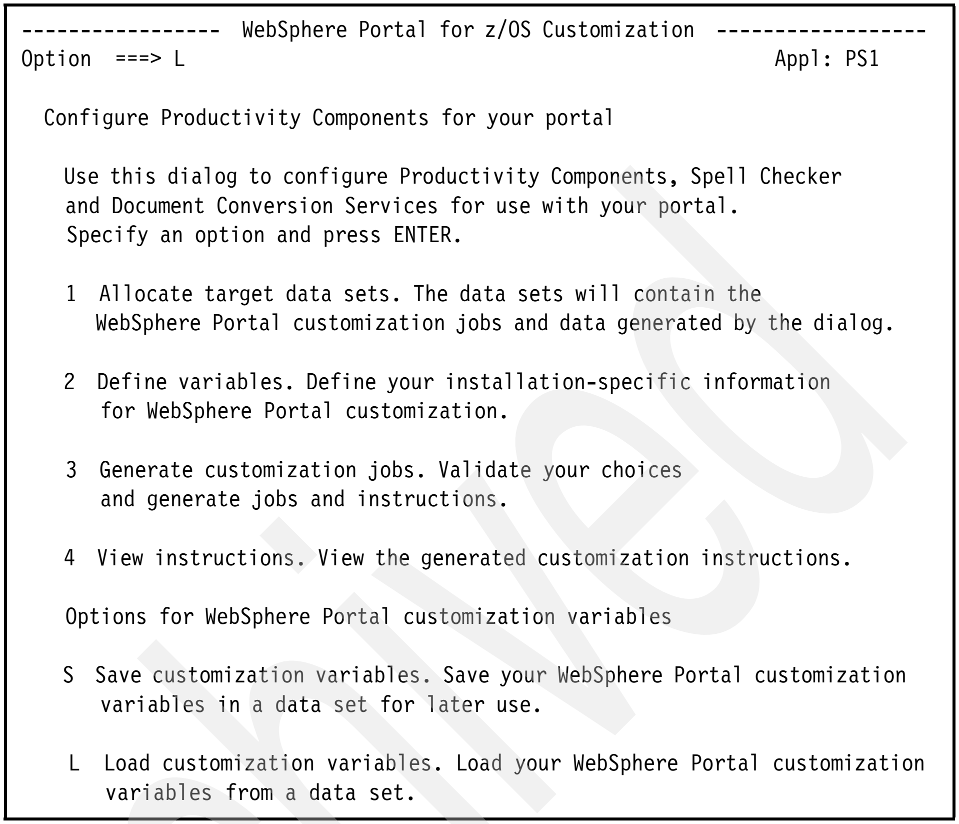


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> 1                                                Appl: PS1

  Configure Productivity Components

    Use this dialog to configure Productivity Components for use
    with your portal. Specify an option and press ENTER.


    1  Configure Productivity Components for your portal.
       If you want to configure productivity portlets, enable remote
       Document Conversion Services, and a remote Spell Checker server
       for your portal, use this option.

    2  Remove Productivity Components from your configuration.
       If you want to disable remote Document Conversion Services and
       the remote Spell Checker server for your portal, use this option.
```

Figure 8-31 Configuring productivity components

Now you need to load the ISPF customization variables that you saved when you last used the Portal customization dialogs.

Load these variables using option **L Load customization variables**, as shown in Figure 8-32 on page 350.

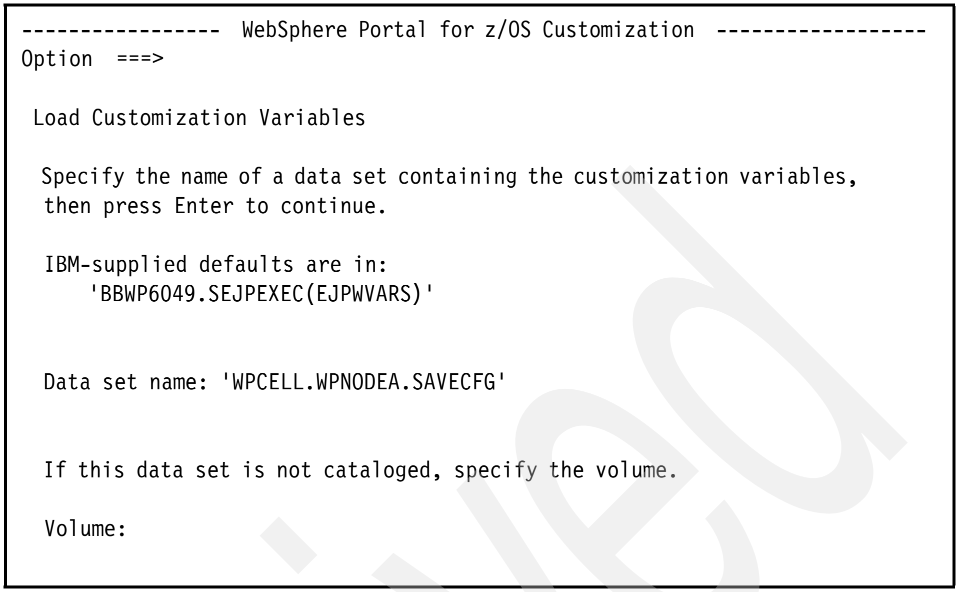


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> L                                                     Appl: PS1

  Configure Productivity Components for your portal

    Use this dialog to configure Productivity Components, Spell Checker
    and Document Conversion Services for use with your portal.
    Specify an option and press ENTER.

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

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

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

    4  View instructions. View the generated customization instructions.

    Options for WebSphere Portal customization variables

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

    L  Load customization variables. Load your WebSphere Portal customization
       variables from a data set.
```

Figure 8-32 Loading customization variables

You are prompted for the location of the data set where you saved the customization variables, as shown in Figure 8-33 on page 351.

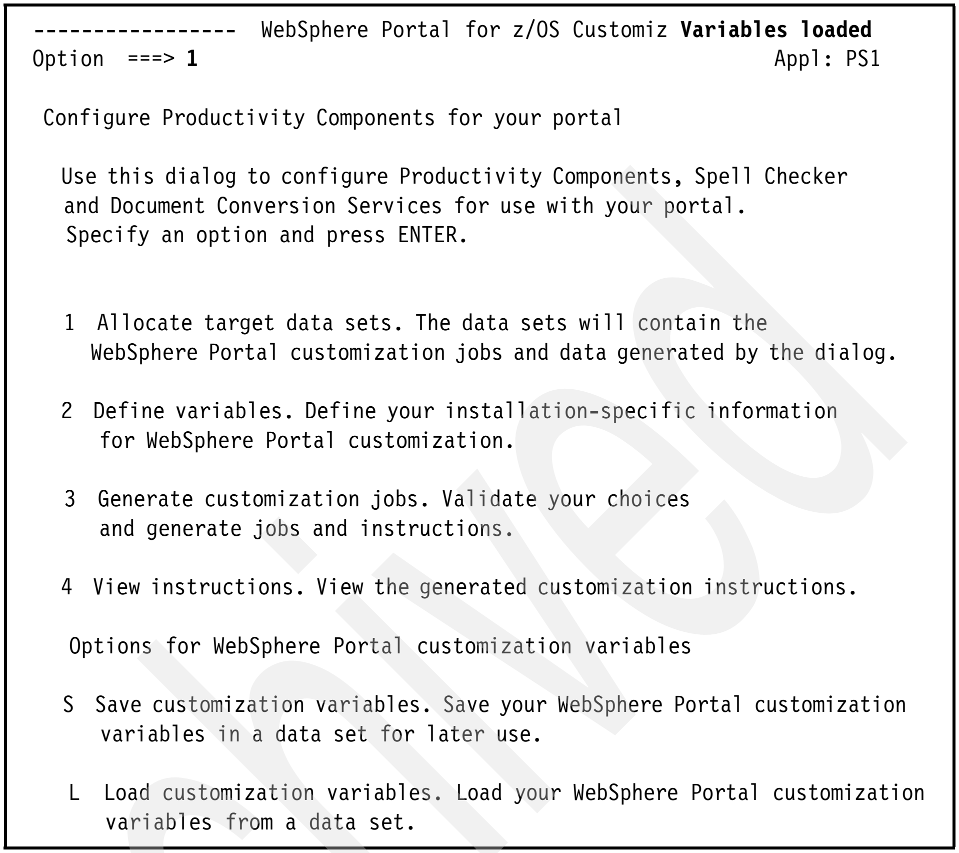
If you have only one set of variables, the screen will displayed populated with the correct data set name. Press Enter to load the customization variables from the specified data set.



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

 Load Customization Variables

  Specify the name of a data set containing the customization variables,
  then press Enter to continue.

  IBM-supplied defaults are in:
      'BBWP6049.SEJPEXEC(EJPWVARS)'


  Data set name: 'WPCELL.WPNODEA.SAVECFG'


  If this data set is not cataloged, specify the volume.

  Volume:
```

Figure 8-33 Specify location of customization variables

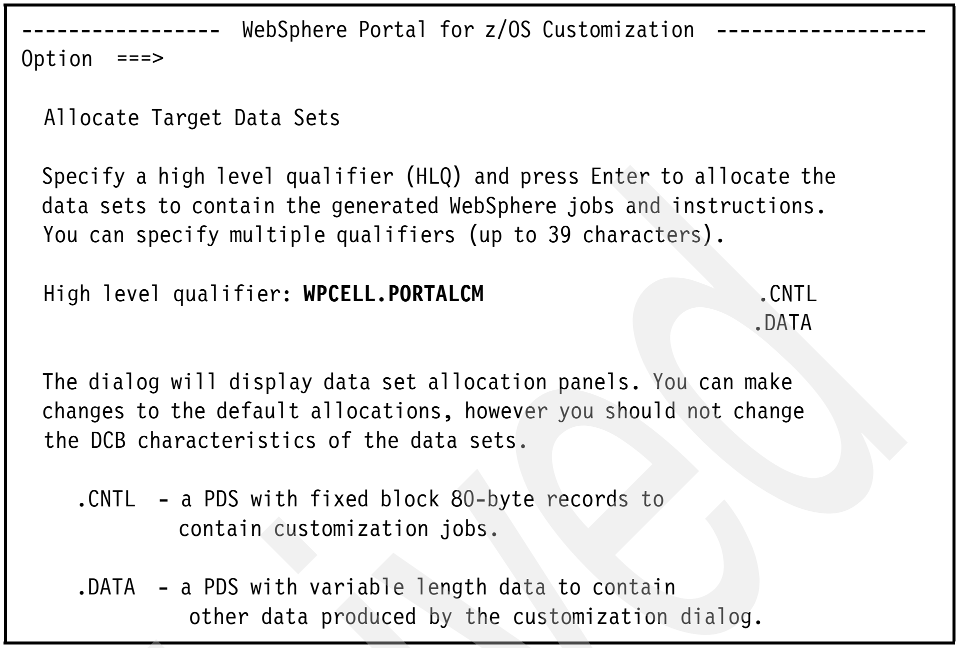
Select option **1 Allocate target data sets** that will be used by the customization panels, as shown in Figure 8-34 on page 352.



```
-----------------  WebSphere Portal for z/OS Customiz Variables loaded
Option  ===> 1                                                 Appl: PS1

 Configure Productivity Components for your portal

   Use this dialog to configure Productivity Components, Spell Checker
   and Document Conversion Services for use with your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.

    Options for WebSphere Portal customization variables

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 8-34 Allocating data sets for customization jobs

You are prompted for a high level qualifier (HLQ) as shown in Figure 8-35 on page 353. In our case, we used `WPCELL.PORTALCM`.

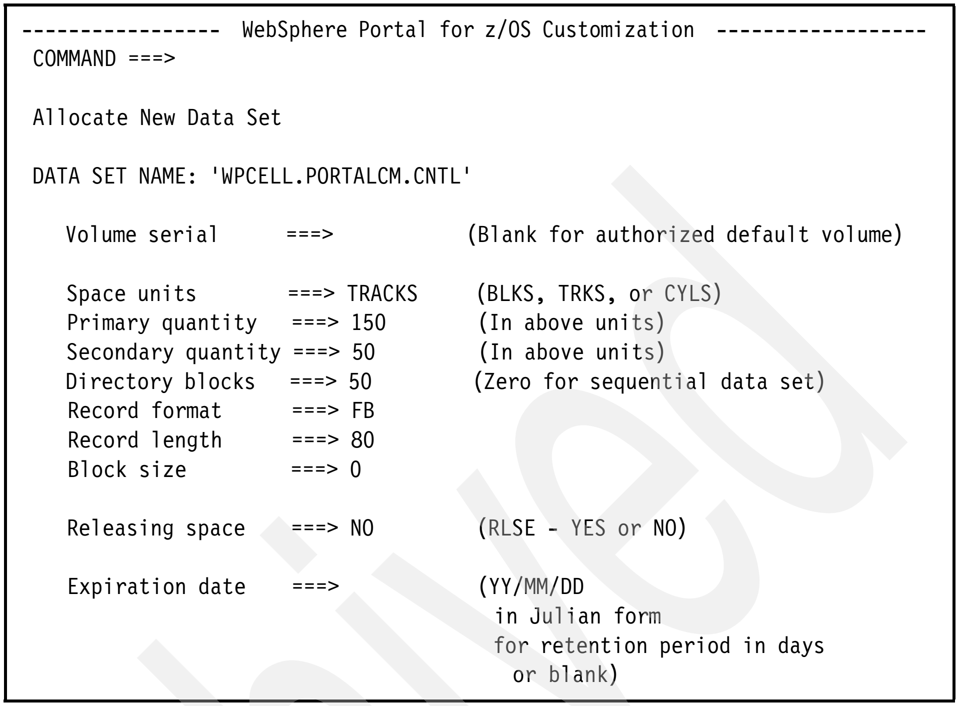


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===>

  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: WPCELL.PORTALCM                        .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 customization jobs.

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

Figure 8-35   Specifying HLQ for customization data sets

Specify the required data set characteristics for the .CNTL and .DATA data sets.

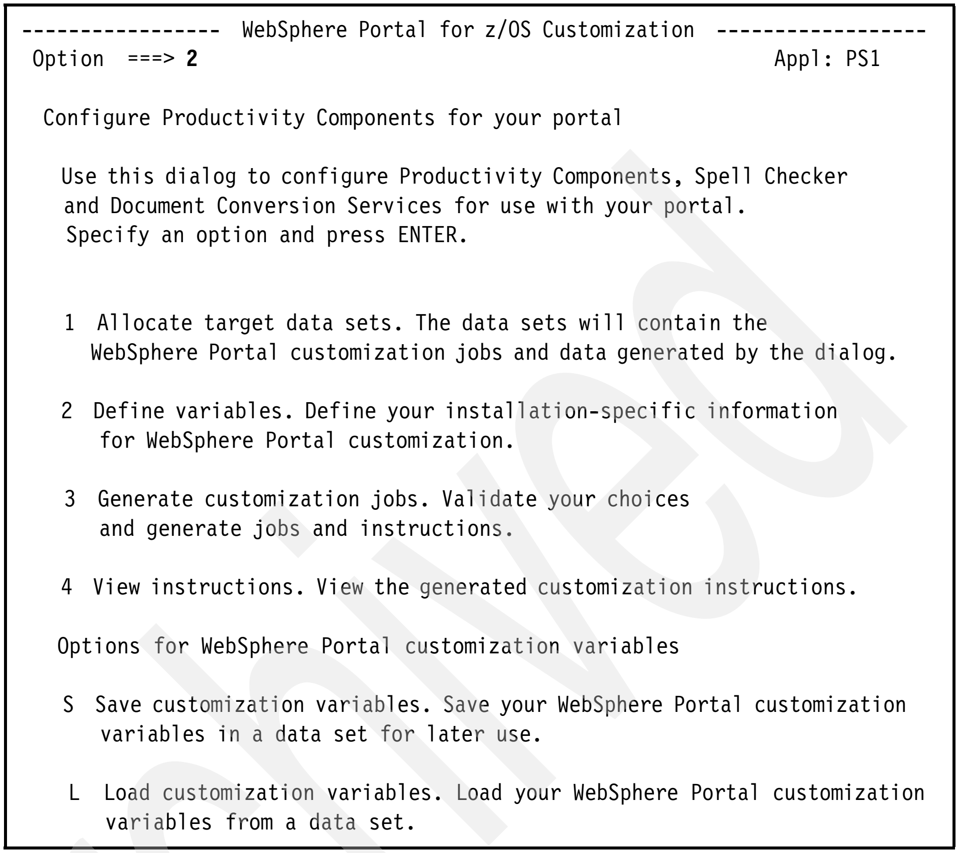
Accept the default allocations in each case, as shown in Figure 8-36 on page 354.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 COMMAND ===>

 Allocate New Data Set

 DATA SET NAME: 'WPCELL.PORTALCM.CNTL'

    Volume serial      ===>           (Blank for authorized default volume)

    Space units        ===> TRACKS     (BLKS, TRKS, or CYLS)
    Primary quantity   ===> 150        (In above units)
    Secondary quantity ===> 50         (In above units)
    Directory blocks   ===> 50        (Zero for sequential data set)
    Record format      ===> FB
    Record length      ===> 80
    Block size         ===> 0

    Releasing space    ===> NO         (RLSE - YES or NO)

    Expiration date    ===>            (YY/MM/DD
                                        in Julian form
                                        for retention period in days
                                          or blank)
```

Figure 8-36 Specifying data set characteristics for customization data sets

Select option **2 Define variables**, as shown in Figure 8-37 on page 355.

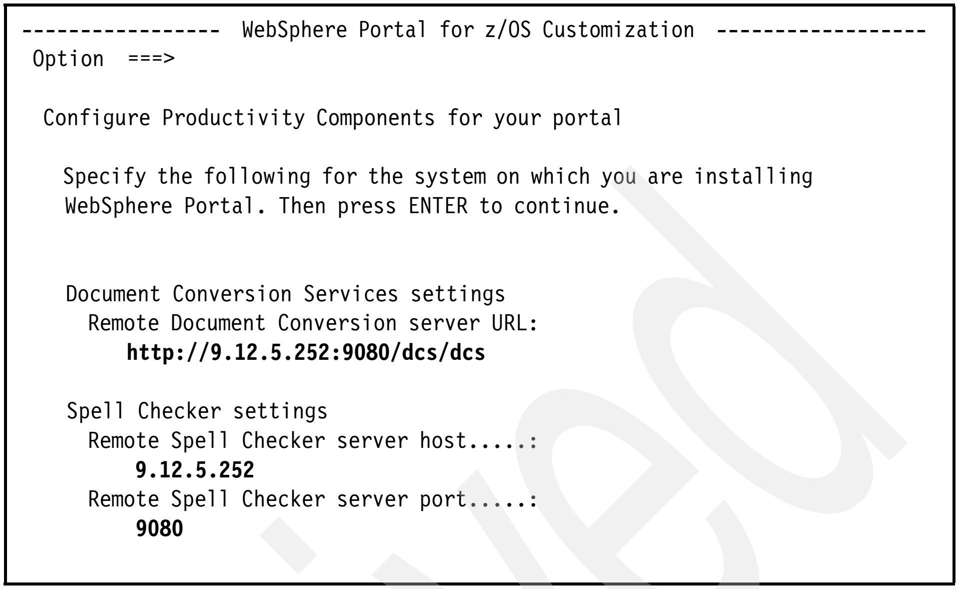


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> 2                                                   Appl: PS1

  Configure Productivity Components for your portal

    Use this dialog to configure Productivity Components, Spell Checker
    and Document Conversion Services for use with your portal.
    Specify an option and press ENTER.


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

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

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

    4  View instructions. View the generated customization instructions.

    Options for WebSphere Portal customization variables

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

    L  Load customization variables. Load your WebSphere Portal customization
       variables from a data set.
```

Figure 8-37 Select menu option to customization variables

You now need to provide the customization dialogs with the information required to access the productivity components on the remote DCS server, as shown in Figure 8-38 on page 356.

> **Note:** In our case, we use the same remote server for both Document Conversion Services (DCS) and spell checking, so for us the host name is the same for both services.
>
> You may choose to use different servers for each of these two services. In this case, the host names you enter in Figure 8-38 for each of these two services would be different.



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

  Configure Productivity Components for your portal

    Specify the following for the system on which you are installing
    WebSphere Portal. Then press ENTER to continue.


    Document Conversion Services settings
      Remote Document Conversion server URL:
         http://9.12.5.252:9080/dcs/dcs

    Spell Checker settings
      Remote Spell Checker server host.....:
          9.12.5.252
      Remote Spell Checker server port.....:
          9080
```

Figure 8-38 Define DCS and Spell Checker configuration

There are three required fields:

- The URL at which the dcs_war application is deployed
- The host name or IP address of the remote Spell Checker
- The port number on which the remote Spell Checker listens

> **Note:** Observe the context root in the Remote Document Conversion server URL.
>
> In 8.3.2, “Deploy the dcs.war file” on page 312, we use just dcs as the context root for the application.
>
> On this panel, however, the URL needs to be specified as `/dcs/dcs`.

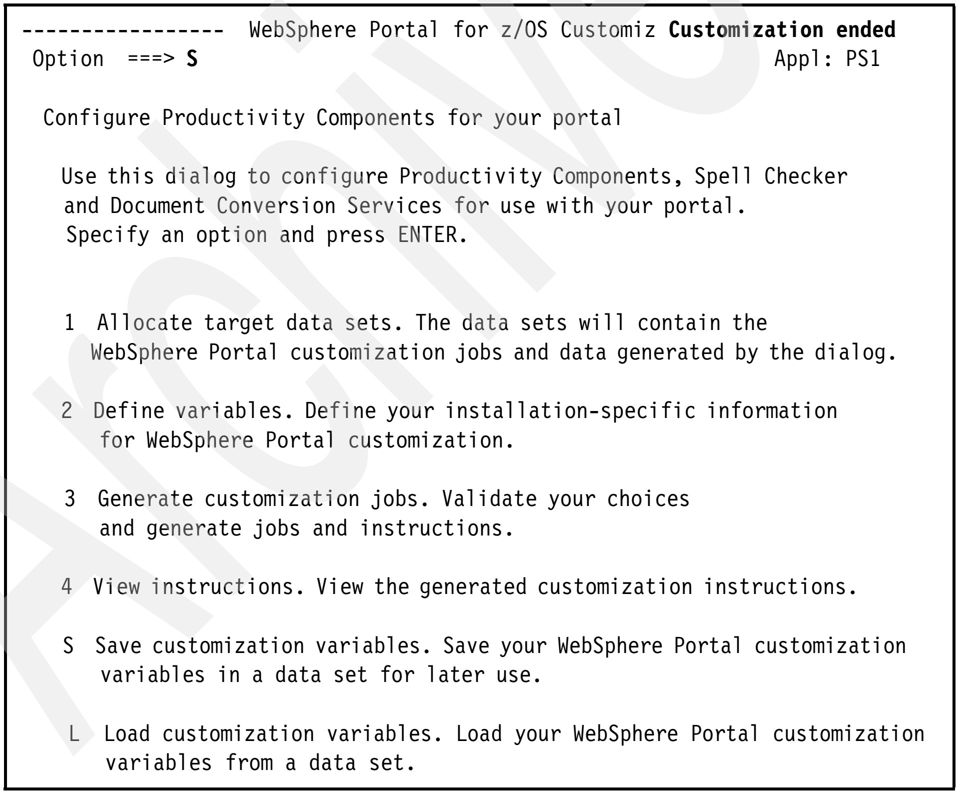
Tip: In our environment, we used the IP address for our remote DCS server because this server was not registered with our Domain Name Server (DNS).

However you would normally use a host name such as dcsserver.yourcompany.com which provides better device independence in case of server relocations.

After you enter the required settings, press Enter.

Now use option **S** to save the customization variables for future use, as shown in Figure 8-39.

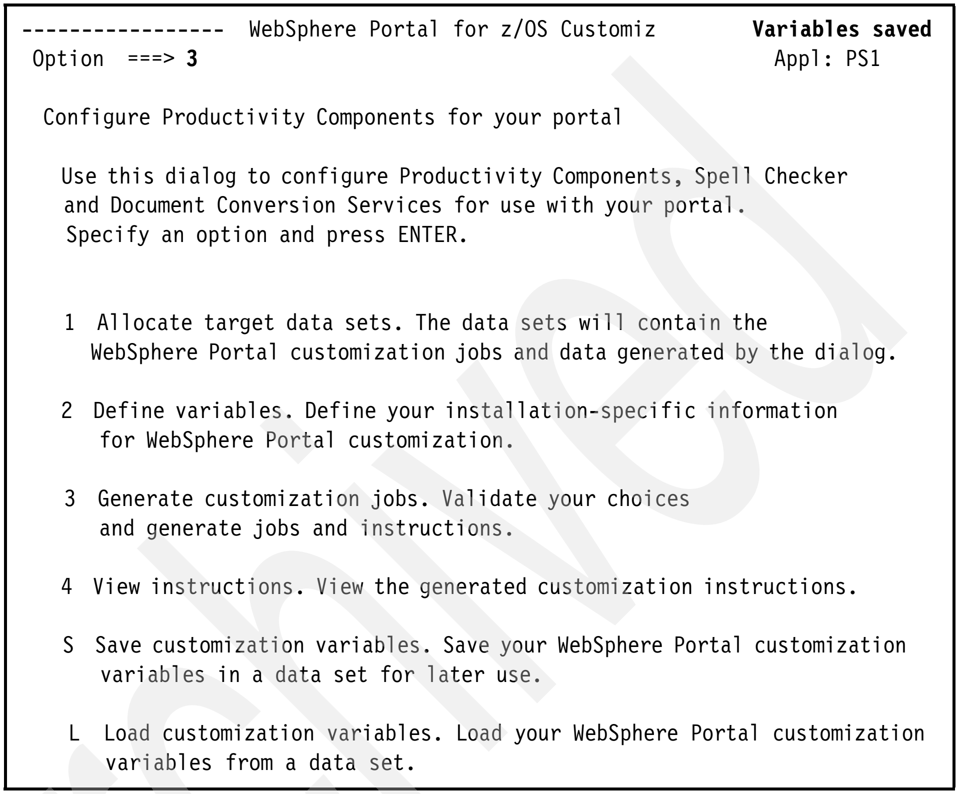
As with loading of the customization variables in Figure 8-33 on page 351, you will be prompted for the location of the data set in which to save the variables. In general, it is easiest to use the same data set each time.



```
-----------------  WebSphere Portal for z/OS Customiz Customization ended
 Option  ===> S                                          Appl: PS1

  Configure Productivity Components for your portal

   Use this dialog to configure Productivity Components, Spell Checker
   and Document Conversion Services for use with your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.

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

    L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

Figure 8-39   Saving the customization variables

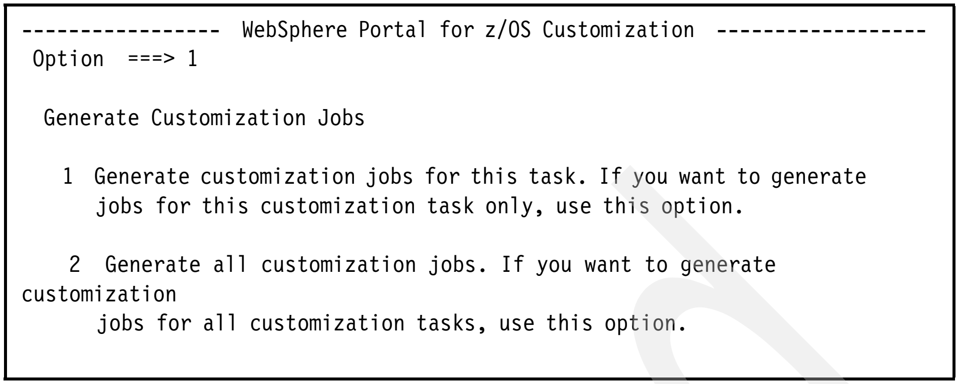
Select option **3** and press Enter as shown in Figure 8-40 to generate the customization jobs which actually perform the Portal configuration for remote DCS services.



```
-----------------  WebSphere Portal for z/OS Customiz       Variables saved
 Option  ===> 3                                               Appl: PS1

  Configure Productivity Components for your portal

    Use this dialog to configure Productivity Components, Spell Checker
    and Document Conversion Services for use with your portal.
    Specify an option and press ENTER.


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

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

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

    4  View instructions. View the generated customization instructions.

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

    L  Load customization variables. Load your WebSphere Portal customization
       variables from a data set.
```

Figure 8-40 Generating customization jobs

On the next screen select option **1 Generate customization jobs for this task** only and press Enter, as shown in Figure 8-41 on page 359.

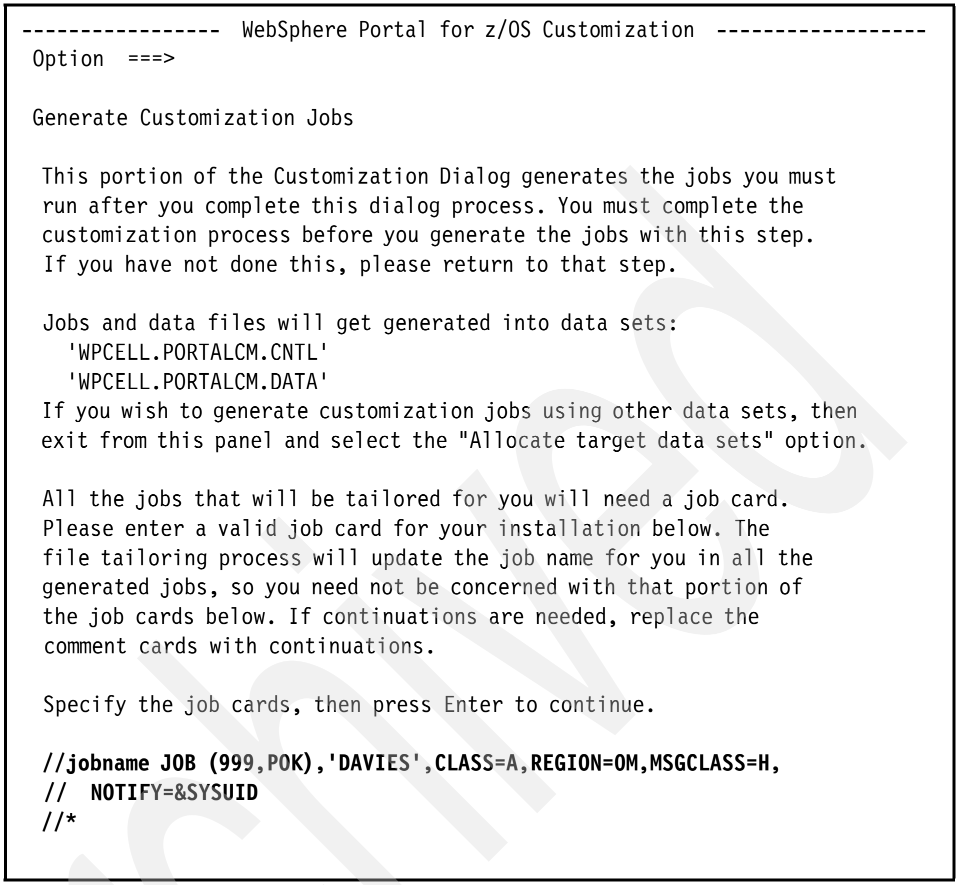


```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> 1

  Generate Customization Jobs

    1  Generate customization jobs for this task. If you want to generate
       jobs for this customization task only, use this option.

    2  Generate all customization jobs. If you want to generate
customization
       jobs for all customization tasks, use this option.
```

Figure 8-41   Generating jobs only for the current task

You are informed where your jobs and data files will be created, and you are prompted for jobcard information as shown in Figure 8-42 on page 360.

When satisfied with your job card information, press Enter.



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

 Generate Customization Jobs

  This portion of the Customization Dialog generates the jobs you must
  run after you complete this dialog process. You must complete the
  customization process before you generate the jobs with this step.
  If you have not done this, please return to that step.

  Jobs and data files will get generated into data sets:
    'WPCELL.PORTALCM.CNTL'
    'WPCELL.PORTALCM.DATA'
  If you wish to generate customization jobs using other data sets, then
  exit from this panel and select the "Allocate target data sets" option.

  All the jobs that will be tailored for you will need a job card.
  Please enter a valid job card for your installation below. The
  file tailoring process will update the job name for you in all the
  generated jobs, so you need not be concerned with that portion of
  the job cards below. If continuations are needed, replace the
  comment cards with continuations.

  Specify the job cards, then press Enter to continue.

  //jobname JOB (999,POK),'DAVIES',CLASS=A,REGION=0M,MSGCLASS=H,
  //  NOTIFY=&SYSUID
  //*
```

Figure 8-42 Customization job tailoring control screen

You will see that your jobs and data files are created, as shown in Figure 8-43 on page 361.

```
Processing for data set 'WPCELL.PORTALCM.CNTL' ...

 Member EJPIDCS successfully created.
 Member EJPSDCSD successfully created.
 Member EJPSSPCD successfully created.

 Processing for data set 'WPCELL.PORTALCM.DATA' ...

 Member EJP2DCSD successfully created.
 Member EJP2SPCD successfully created.

 ***
```

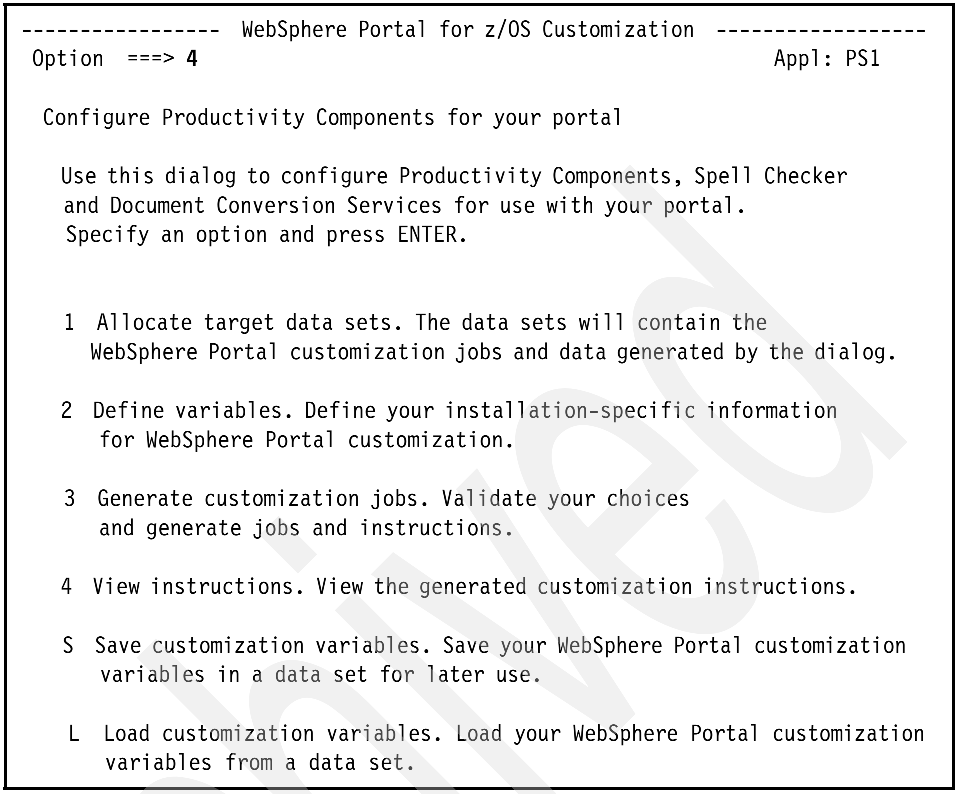
Figure 8-43 Successful creation of jobs and data files

Table 8-3 lists the descriptions of the generated members of the .CNTL library.

Table 8-3 *Description of generated `.CNTL` members.*

Generated job	Description
EJPIDCS	This job contains the generated instructions.
EJPSDCSD	This job is used to delegate Document Conversions to an external WebSphere Application Server host.
EJPSSPCD	This job is used to delegate the Spell Checker function to an external WebSphere Application Server host.

On the screen Configure Productivity Components for your portal, use option **4 View instructions** to see the instructions for the generated jobs, as shown in Figure 8-44 on page 362.



```
----------------  WebSphere Portal for z/OS Customization  -----------------
 Option  ===> 4                                                 Appl: PS1

  Configure Productivity Components for your portal

   Use this dialog to configure Productivity Components, Spell Checker
   and Document Conversion Services for use with your portal.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.

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

    L  Load customization variables. Load your WebSphere Portal customization
       variables from a data set.
```

Figure 8-44   Viewing the instructions for the customization jobs

This option simply invokes the ISPF view command for the member EJPIDCS. Follow these instructions when running the jobs. The instructions also provide references to the Portal for z/OS Information Center available on the Internet. These references describe prerequisite tasks described in 8.3, “Configuring DCS and Spell Checker on the remote server” on page 310.

> **Note:** We recommend that you read the Information Center references and satisfy yourself that all prerequisite tasks have been completed, in case other prerequisite tasks are added to the Information Center topics subsequent to the publication of this book.

When you have completed all prerequisite tasks listed in the instructions in member EJPIDCS, run each of the two required jobs, EJPSDCSD and EJPSSPCD in the sequence described in those instructions and check the output after each job before proceeding to the next job.

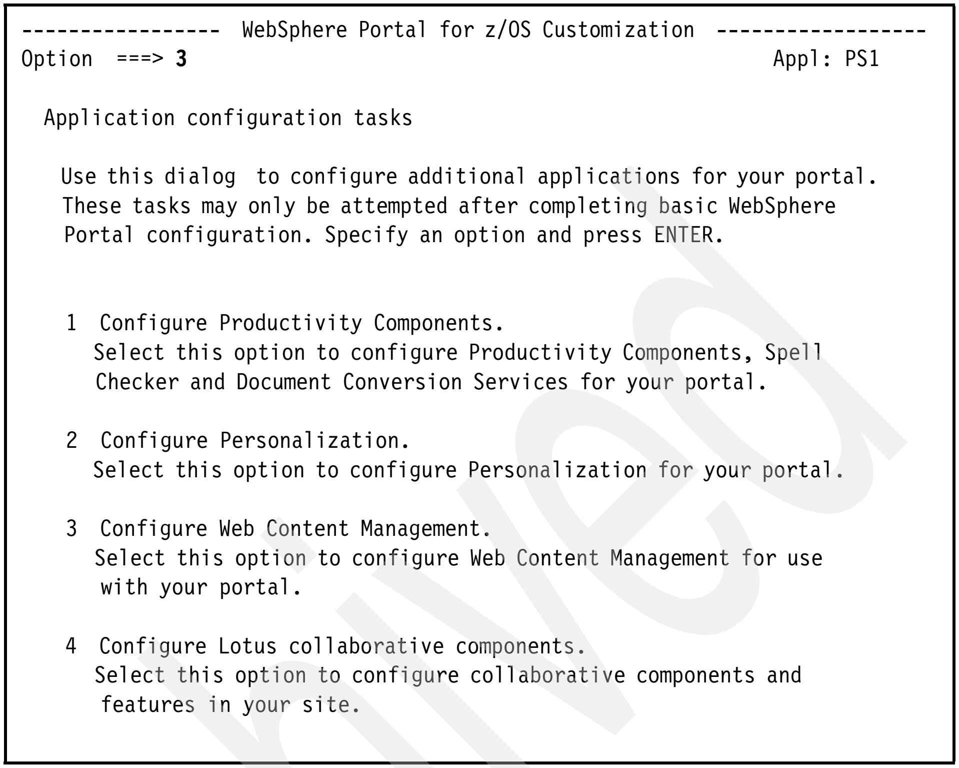
You should have zero (0) return codes for each job step. However, the instructions also recommend that you visually inspect the job output further to look for any messages indicating failure.

> **Note:** We do *not* recommend running these Portal customization jobs at the same time as other Portal customization jobs described in other sections of this book.
>
> Running concurrent customization jobs can lead to conflicts in the execution of these jobs.

8.4.3 Deploying authoring portlets

To deploy the authoring portlets, you use the same customization dialog as used in 8.4.2, “Configuring the Portal server to use remote DCS services” on page 344.

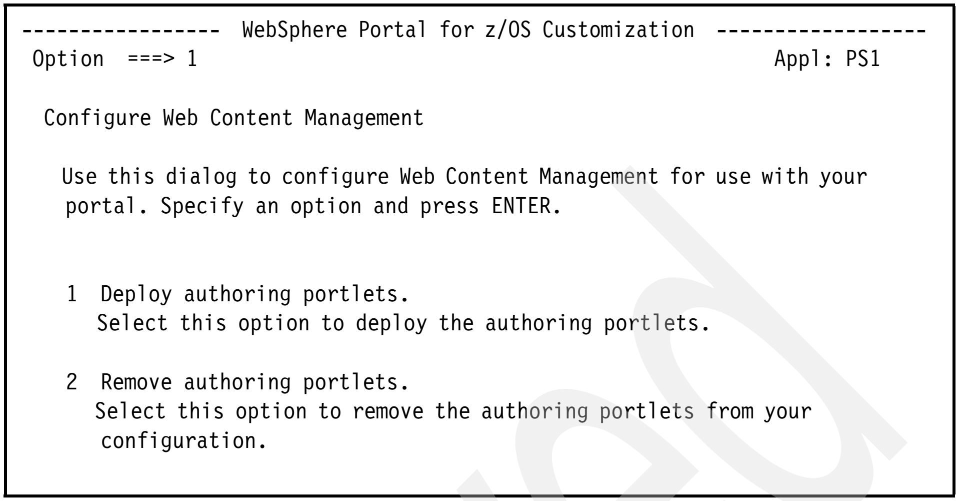
Start the dialog as before and navigate to the screen Application configuration tasks, as shown in Figure 8-45 on page 364. Choose option **3 Configure Web Content Management**.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
Option  ===> 3                                                      Appl: PS1

 Application configuration tasks

   Use this dialog  to configure additional applications for your portal.
   These tasks may only be attempted after completing basic WebSphere
   Portal configuration. Specify an option and press ENTER.


   1  Configure Productivity Components.
      Select this option to configure Productivity Components, Spell
      Checker and Document Conversion Services for your portal.

   2  Configure Personalization.
      Select this option to configure Personalization for your portal.

   3  Configure Web Content Management.
      Select this option to configure Web Content Management for use
      with your portal.

   4  Configure Lotus collaborative components.
      Select this option to configure collaborative components and
      features in your site.
```

Figure 8-45 Configuring Web Content Management

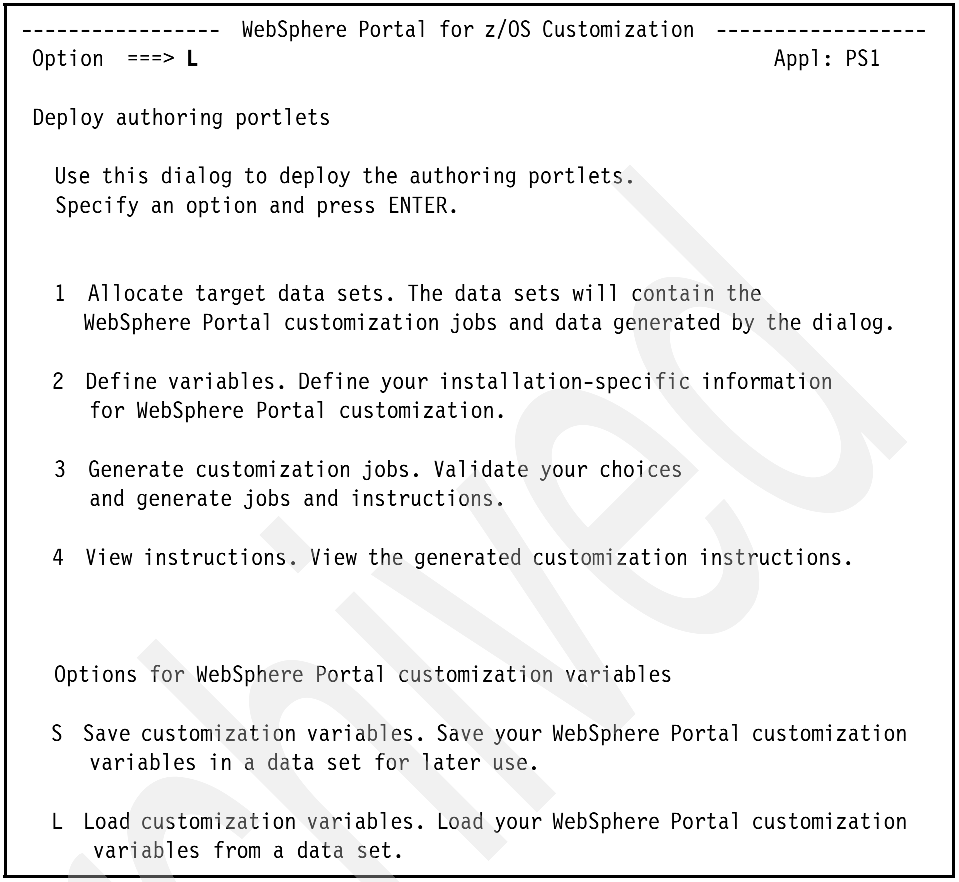
On the screen Configure Web Content Management, as shown in Figure 8-46 on page 365, select option **1 Deploy authoring portlets**.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> 1                                                 Appl: PS1

  Configure Web Content Management

    Use this dialog to configure Web Content Management for use with your
    portal. Specify an option and press ENTER.


    1  Deploy authoring portlets.
       Select this option to deploy the authoring portlets.

    2  Remove authoring portlets.
       Select this option to remove the authoring portlets from your
       configuration.
```

Figure 8-46 Selecting Deploy authoring portlets

Use option **L Load customization variables** to load the previously saved ISPF customization variables on the screen Deploy authoring portlets, as shown in Figure 8-47 on page 366.



```
-----------------  WebSphere Portal for z/OS Customization  ------------------
 Option  ===> L                                                   Appl: PS1

 Deploy authoring portlets

   Use this dialog to deploy the authoring portlets.
   Specify an option and press ENTER.


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

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

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

   4  View instructions. View the generated customization instructions.



   Options for WebSphere Portal customization variables

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

   L  Load customization variables. Load your WebSphere Portal customization
      variables from a data set.
```

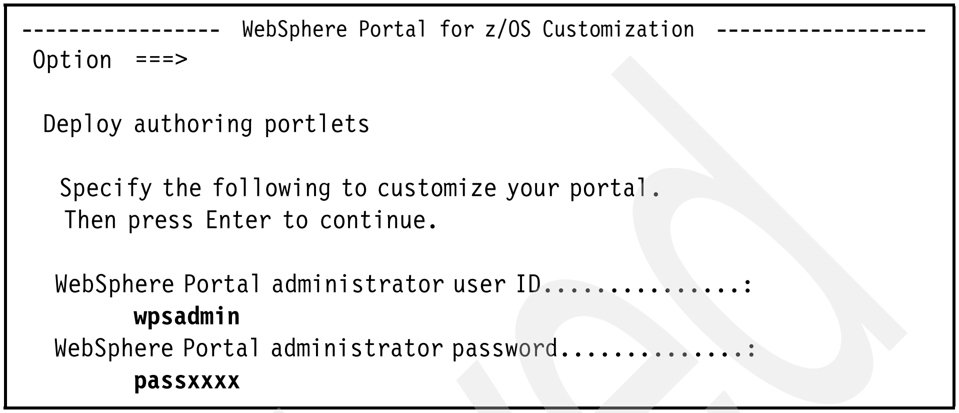
Figure 8-47 Loading customization variables for authoring portlet deployment

As with Figure 8-32 on page 350, you will be prompted for the location of the data set where you last saved the customization variables. The screen will be prepopulated with the last data set name you use.

After loading the customization variables, use option **1 Allocate target data sets** on the same screen Deploy authoring portlets as shown in Figure 8-47. (We recommend that you use different data sets for this task from those used in Figure 8-34 on page 352.)

Use option **2 Define variables**, from the same screen shown in Figure 8-47, to define your variables.

Next, on the screen Deploy authoring portlets as shown in Figure 8-48, type the user ID and password for the WebSphere Portal administrative user ID and press Enter.



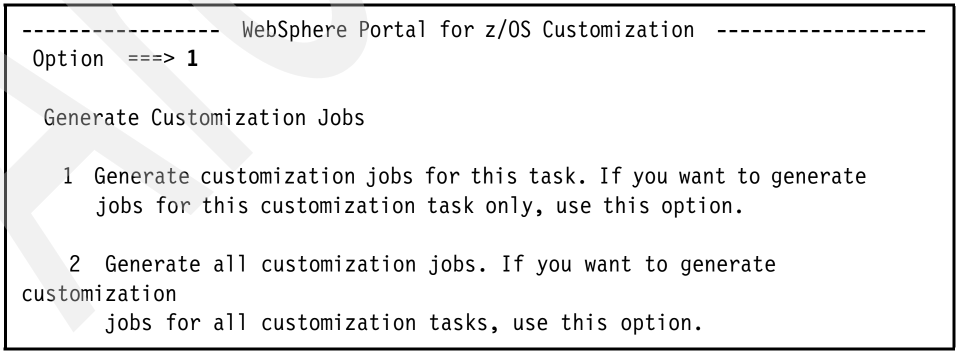
```
----------------  WebSphere Portal for z/OS Customization  -----------------
 Option  ===>

  Deploy authoring portlets

    Specify the following to customize your portal.
     Then press Enter to continue.

   WebSphere Portal administrator user ID..............:
          **wpsadmin**
   WebSphere Portal administrator password.............:
          **passxxxx**
```

Figure 8-48   Specifying the Portal administrator user ID and password

Save the ISPF customization variables using option **S** from the screen Deploy authoring portlets as shown in Figure 8-47 on page 366. You will be prompted again for the data set to save the variables and we recommend that you use the same data set from which you loaded the variables.

On the screen Deploy authoring portlets, as shown in Figure 8-47 on page 366, select option **3 Generate customization jobs**.

On the next screen, select option **1 Generate customization jobs for this task** only and press Enter, as shown in Figure 8-49.

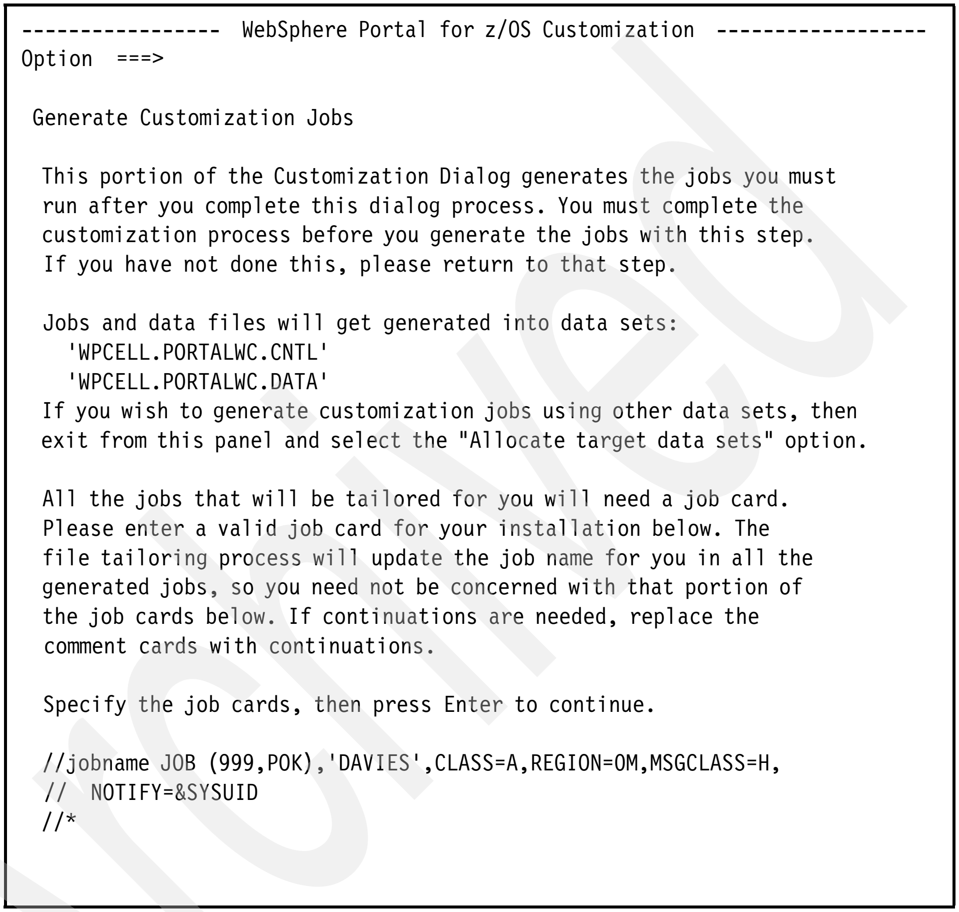


```
----------------  WebSphere Portal for z/OS Customization  -----------------
 Option  ===> 1

  Generate Customization Jobs

    1  Generate customization jobs for this task. If you want to generate
       jobs for this customization task only, use this option.

     2  Generate all customization jobs. If you want to generate
customization
        jobs for all customization tasks, use this option.
```

Figure 8-49   Generating jobs for the current task only

You are informed where your jobs and data files will be created, and you are prompted for jobcard information as shown in Figure 8-50.

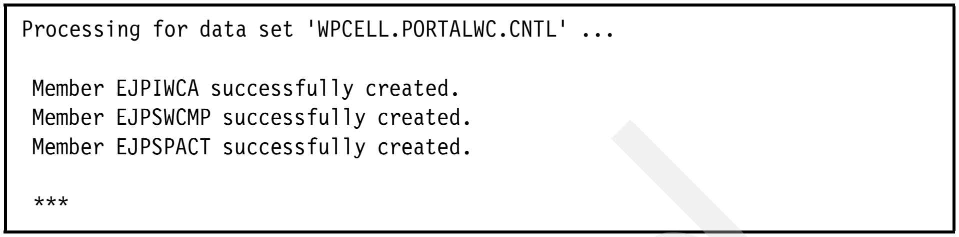
When satisfied with your job card information, press Enter.



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

 Generate Customization Jobs

  This portion of the Customization Dialog generates the jobs you must
  run after you complete this dialog process. You must complete the
  customization process before you generate the jobs with this step.
  If you have not done this, please return to that step.

  Jobs and data files will get generated into data sets:
    'WPCELL.PORTALWC.CNTL'
    'WPCELL.PORTALWC.DATA'
  If you wish to generate customization jobs using other data sets, then
  exit from this panel and select the "Allocate target data sets" option.

  All the jobs that will be tailored for you will need a job card.
  Please enter a valid job card for your installation below. The
  file tailoring process will update the job name for you in all the
  generated jobs, so you need not be concerned with that portion of
  the job cards below. If continuations are needed, replace the
  comment cards with continuations.

  Specify the job cards, then press Enter to continue.

  //jobname JOB (999,POK),'DAVIES',CLASS=A,REGION=0M,MSGCLASS=H,
  //  NOTIFY=&SYSUID
  //*
```

Figure 8-50 Customization job tailoring control screen

You will see that your jobs and data files are created, as shown in Figure 8-51.



```
Processing for data set 'WPCELL.PORTALWC.CNTL' ...

 Member EJPIWCA successfully created.
 Member EJPSWCMP successfully created.
 Member EJPSPACT successfully created.

 ***
```

Figure 8-51 Successful creation of jobs and data files

> **Note:** Currently no data files are created by this step in Portal customization, so the .DATA data set is not used.

Table 8-4 lists the descriptions of the generated members of the .CNTL library.

Table 8-4 *Description of generated `.CNTL` members*

Generated job	Description
EJPIWCA	This job contains the generated instructions.
EJPSWCMP	This job is used to deploy the Web Content Management authoring portlets.
EJPSPACT	This job activates all portlets that have been deployed on your portal server. This is required only if running on the primary node of a Network Deployment cell.

Now you can use option **4 View instructions** from the screen Deploy authoring portlets, as shown in Figure 8-47 on page 366, to see the instructions for the generated jobs.This option simply invokes the ISPF view command for the member EJPIWCA.

Follow these instructions when running the jobs. The instructions also provide references to the Portal for z/OS Information Center available on the Internet. These references describe prerequisite tasks which should already be covered in 8.3, “Configuring DCS and Spell Checker on the remote server” on page 310.

Note: We recommend that you still read these Information Center references and satisfy yourself that all prerequisite tasks have been completed, in case other prerequisite tasks are added to the Information Center topics subsequent to the publication of this book.

After you have completed all prerequisite tasks listed in the instructions in member EJPIWCA, run each of the two required jobs, EJPSWCMP and EJPSPACT in the sequence described in those instructions and check the output after each job before proceeding to the next job. You should have zero (0) return codes for each job step. However, the instructions also recommend that you visually inspect the job output further to look for any messages indicating failure.

Note: We do *not* recommend running these Portal customization jobs at the same time as other Portal customization jobs described in other sections of this book. Running concurrent customization jobs can lead to conflicts in the execution of these jobs.

8.4.4 Verifying the WCM configuration

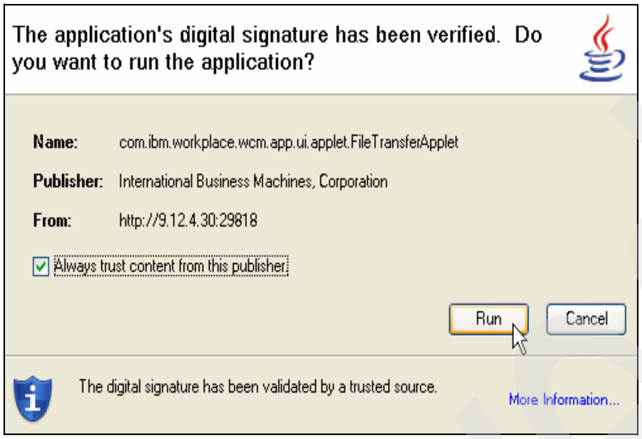
This section describes how to verify that WCM has been configured properly. First we verify that the WCM Authoring Portlet has been deployed. Then we will verify that WebSphere Portal on z/OS is using remote DCS and spell checking by performing these tasks:

- Importing a document into Portal Document Manager (PDM), and previewing it
- Creating a document using the Productive portlets, and spell checking the document

Follow these steps:

1. Use a Web browser to access the Portal login page and log in as the Portal administrator.
2. To access the Web Content page, either click **Launch** in the top left corner and then select **Web Content** from the menu, or click **Web Content** under **Product Links portlet** on the right side.
3. On the Web Content page, find the tabs Web Content Management and Content Preview.
4. The Web Content Management tab is the WCM Authoring portlet. Click the **Web Content Management** tab. If presented with a Warning Security

message, as shown in Figure 8-52, click **Always trust content from this publisher** and then click **Run**.


The application's digital signature has been verified. Do you want to run the application?

Name: com.ibm.workplace.wcm.app.ui.applet.FileTransferApplet

Publisher: International Business Machines, Corporation

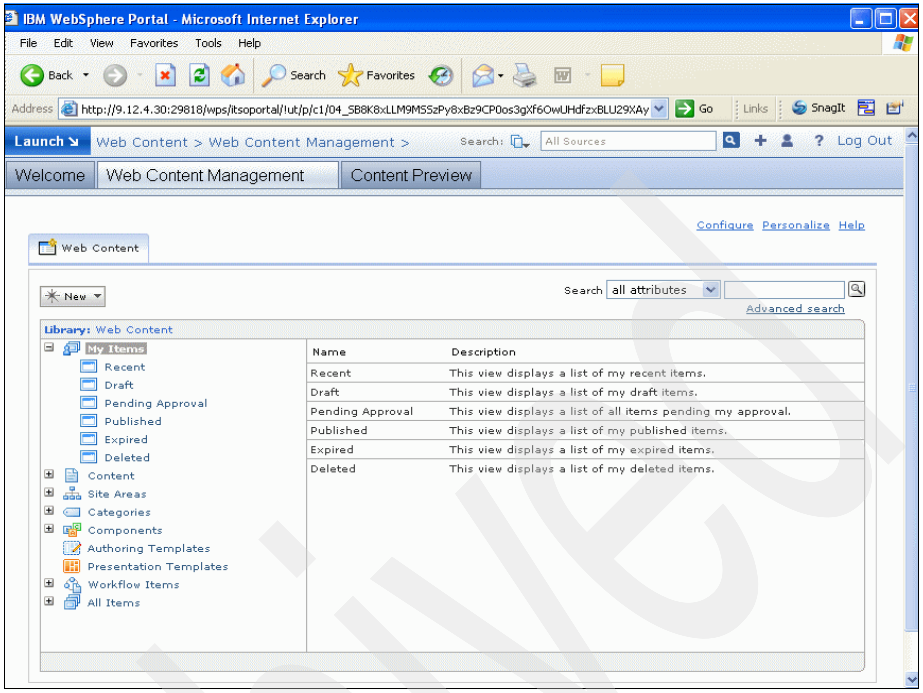
From: http://9.12.4.30:29818

☑ Always trust content from this publisher.

The digital signature has been validated by a trusted source.

Figure 8-52 WCM authoring portlet security warning

5. Figure 8-53 on page 372 is the WCM Authoring Portlet. If the WCM Authoring Portlet is not displayed, then review the configuration steps.



Name	Description
Recent	This view displays a list of my recent items.
Draft	This view displays a list of my draft items.
Pending Approval	This view displays a list of all items pending my approval.
Published	This view displays a list of my published items.
Expired	This view displays a list of my expired items.
Deleted	This view displays a list of my deleted items.

Figure 8-53 WCM Authoring portlet

6. Click **Launch -> Documents**, as shown in Figure 8-54 on page 373, and the PDM welcome page will display.

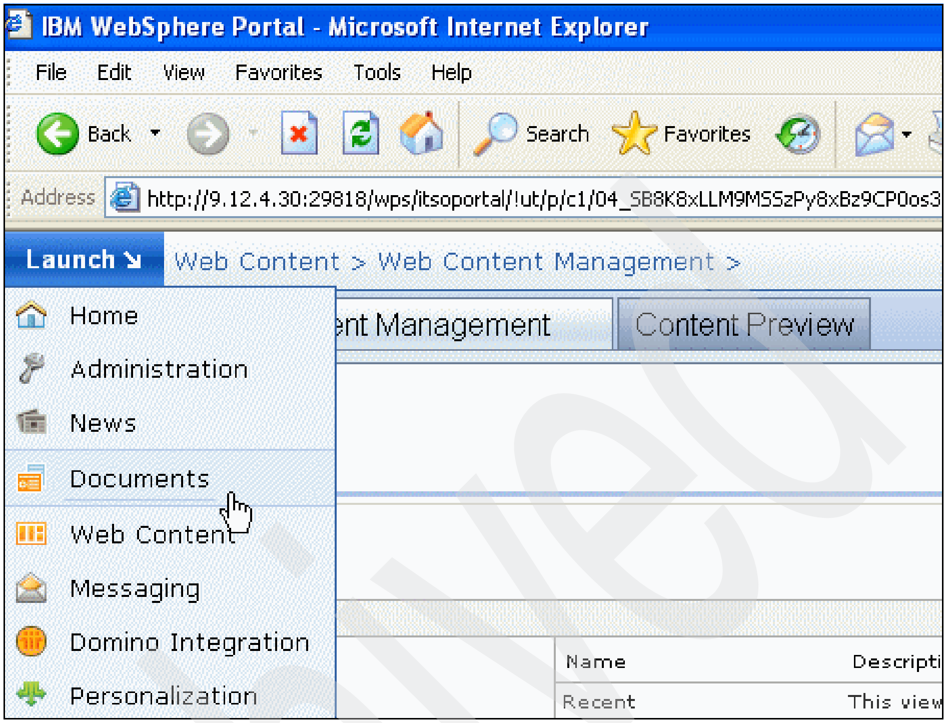
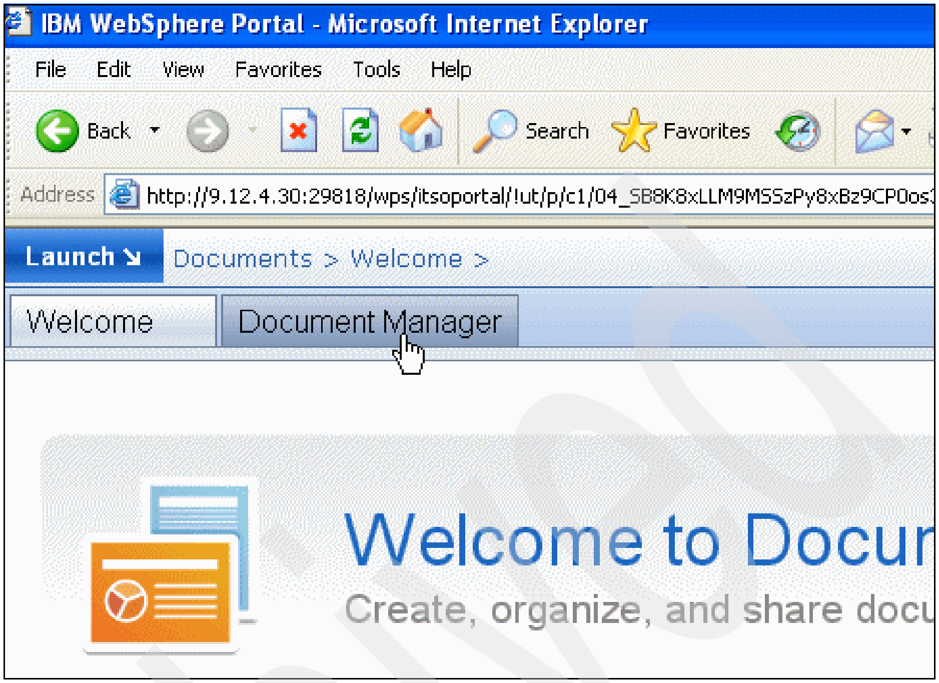


Figure 8-54 Selecting Documents

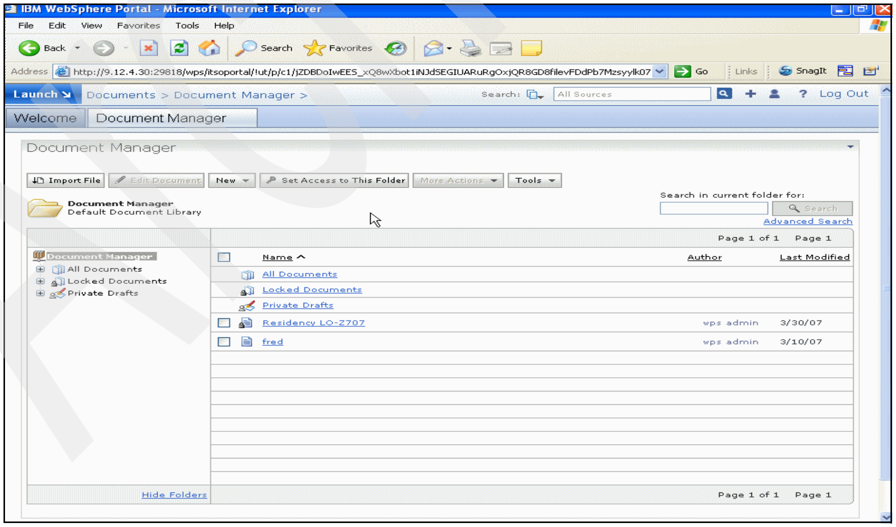
7. Click the **Document Manager** tab next to the Welcome tab, as shown in Figure 8-55 on page 374.



Address: http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os

Figure 8-55   Selecting the Document Manager tab

8. The PDM page will display, as shown in Figure 8-56.



Address: http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jZDBDoIwEES_xQ8wXbot1iNJdSEGIUARuRgOxjQR8GD8filevFDdPb7Mzsyylk07

Document Manager
Default Document Library

	Name ^	Author	Last Modified
	All Documents		
	Locked Documents		
	Private Drafts		
□	Residency LO-Z707	wps admin	3/30/07
□	fred	wps admin	3/10/07

Figure 8-56   PDM portlet

9. Click **Import File**, as shown in Figure 8-57, to import any type of file from your desktop.

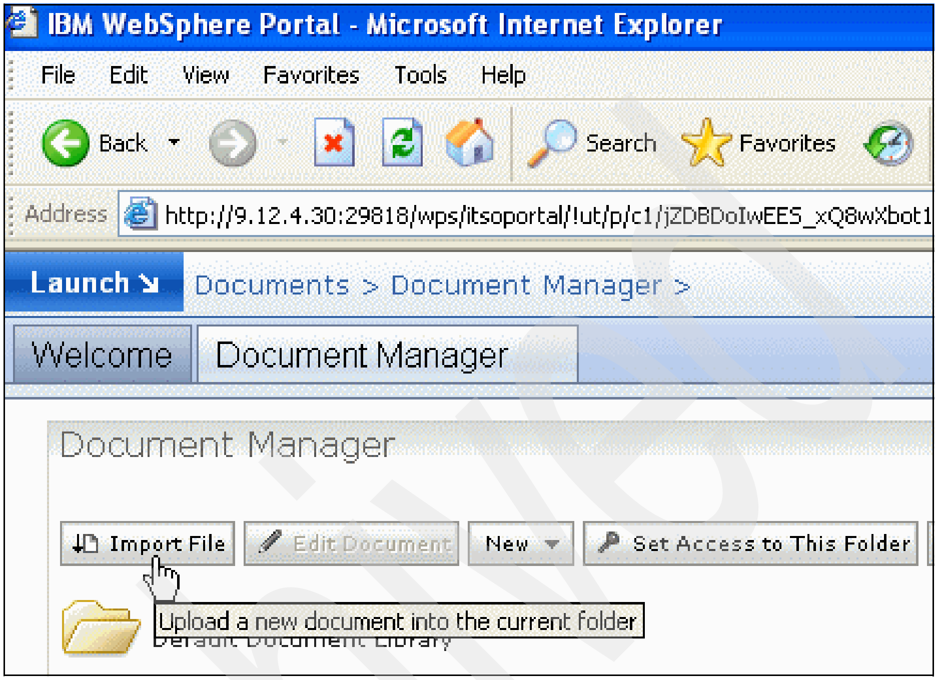
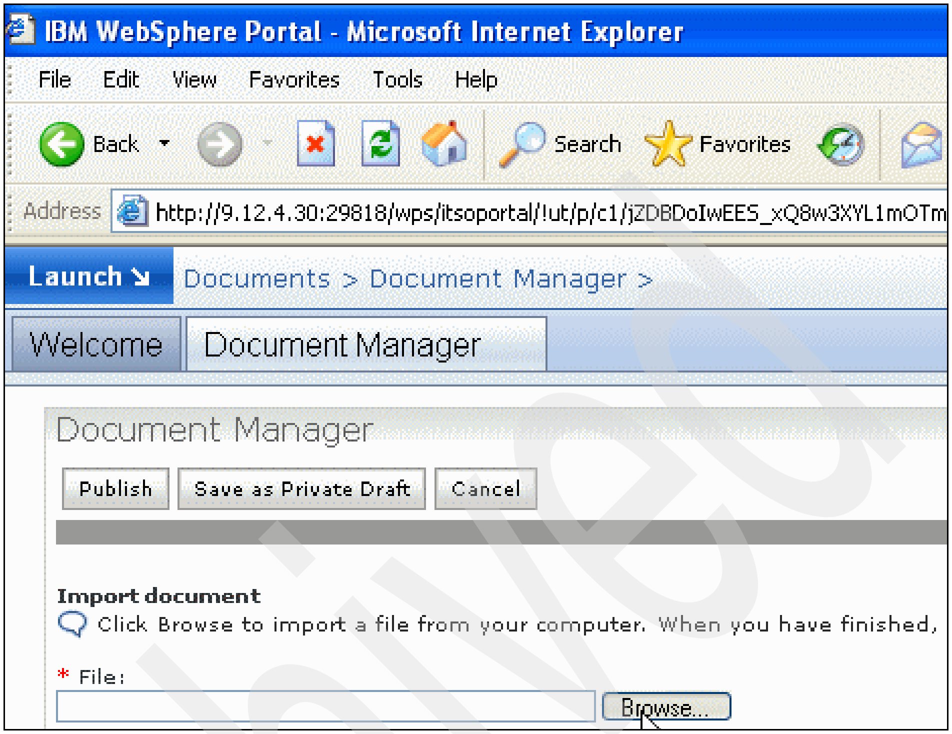

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jZDBDoIwEES_xQ8wXbot1
Launch
Documents > Document Manager >
Welcome
Document Manager
Document Manager
Import File
Edit Document
New
Set Access to This Folder
Upload a new document into the current folder
Default Document Library

Figure 8-57 Selecting Import File

10. Click **Browse...** to select a file from the file system, as shown in Figure 8-58 on page 376.


IBM WebSphere Portal - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jZDBDoIwEES_xQ8w3XYL1mOTm
Launch
Documents > Document Manager >
Welcome
Document Manager
Document Manager
Publish
Save as Private Draft
Cancel
Browse...

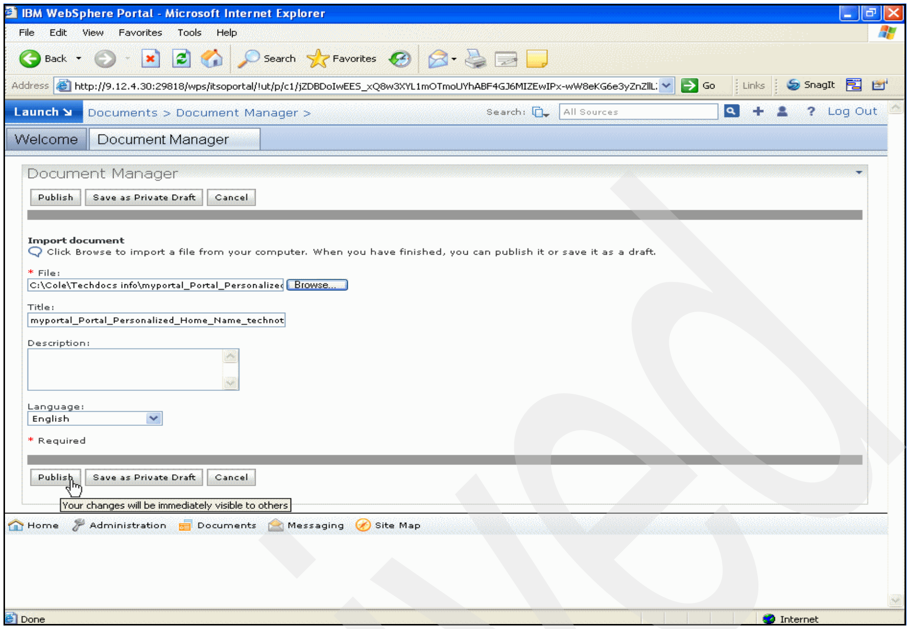
Import document

Click Browse to import a file from your computer. When you have finished,

* File:

Figure 8-58 Selecting Browse... to see the filesystem

11.Click **Publish** to make this document immediately visible, as shown in Figure 8-59 on page 377.



Document Manager

Import document

Click Browse to import a file from your computer. When you have finished, you can publish it or save it as a draft.

* File: C:\Cole\Techdocs info\myportal_Portal_Personalizec

Title: myportal_Portal_Personalized_Home_Name_technot

Description:

Language: English

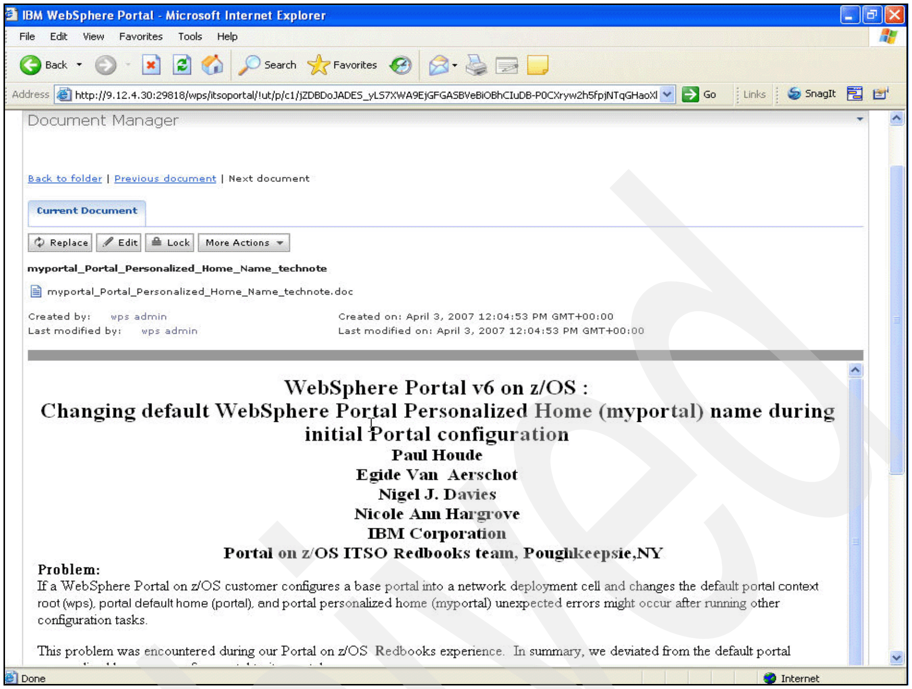
* Required

Your changes will be immediately visible to others

Figure 8-59 Publishing the imported document

12. You should now be back at the PDM portlet initial page, as shown in Figure 8-56 on page 374, and your document should appear in the list.

13. Click your document's name to see a preview of your document that is similar to Figure 8-60 on page 378.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jZDBDoJADES_yLS7XWA9EjGFGASBVeBiOBhCIuDB-P0CXryw2h5fpjNTqGHaoXl
Go
Links
SnagIt
Document Manager
Back to folder | Previous document | Next document
Current Document
Replace
Edit
Lock
More Actions
Done
Internet

myportal_Portal_Personalized_Home_Name_technote

myportal_Portal_Personalized_Home_Name_technote.doc

Created by: wps admin
Last modified by: wps admin

Created on: April 3, 2007 12:04:53 PM GMT+00:00
Last modified on: April 3, 2007 12:04:53 PM GMT+00:00

WebSphere Portal v6 on z/OS :
Changing default WebSphere Portal Personalized Home (myportal) name during initial Portal configuration
Paul Houde
Egide Van Aerschot
Nigel J. Davies
Nicole Ann Hargrove
IBM Corporation
Portal on z/OS ITSO Redbooks team, Poughkeepsie,NY

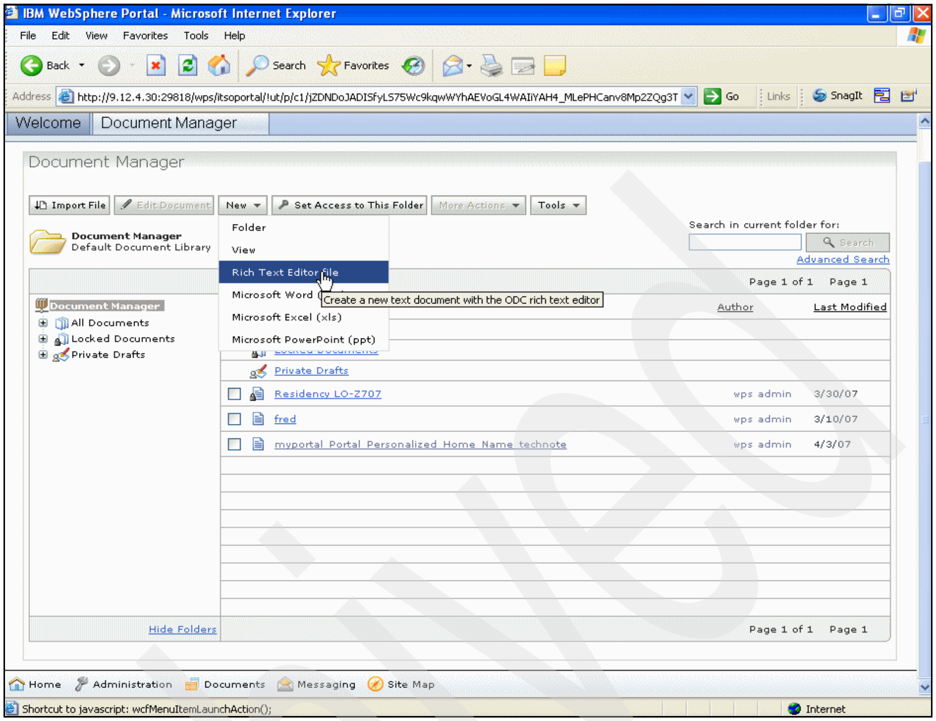
Problem:
If a WebSphere Portal on z/OS customer configures a base portal into a network deployment cell and changes the default portal context root (wps), portal default home (portal), and portal personalized home (myportal) unexpected errors might occur after running other configuration tasks.

This problem was encountered during our Portal on z/OS Redbooks experience. In summary, we deviated from the default portal

Figure 8-60 Previewing the imported document

14.Click **Back** to folder as shown in Figure 8-56 on page 374 to return to the initial PDM page.

15.To create a new document using the Productivity portlets, click **New** -> **Rich Text Editor File**, as shown in Figure 8-61 on page 379.



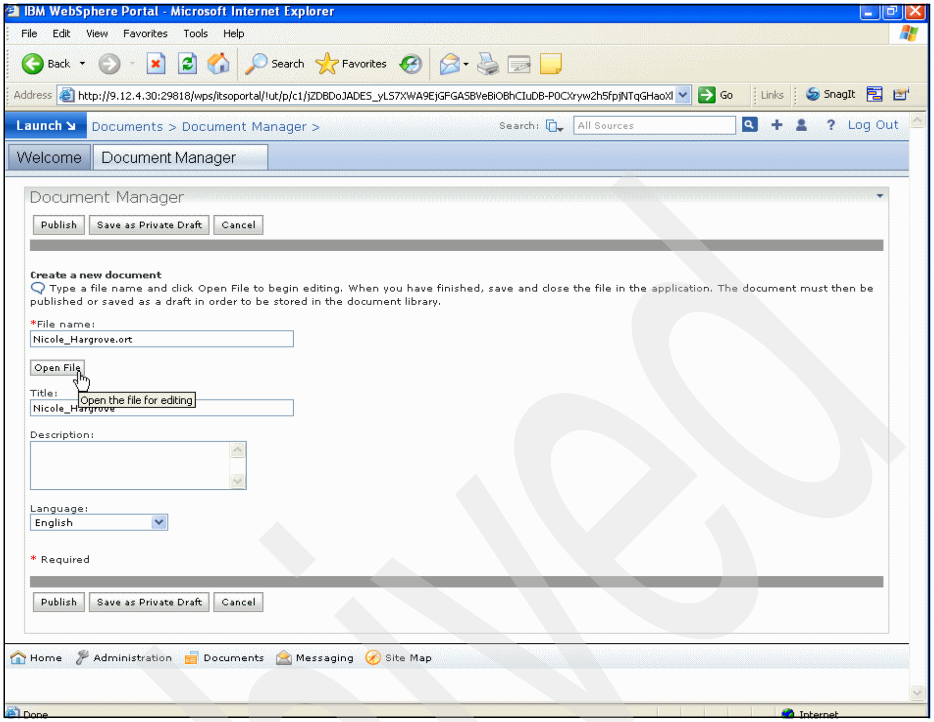
Create a new text document with the ODC rich text editor

	Author	Last Modified
Residency LO-Z707	wps admin	3/30/07
fred	wps admin	3/10/07
myportal Portal Personalized Home Name technote	wps admin	4/3/07

Shortcut to javascript: wcfMenuItemLaunchAction();

Figure 8-61 Selecting Rich Text editor

16. Add a file name to the .ort extension and click **Open File**, as shown in Figure 8-62 on page 380.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jZDBDoJADES_yLS7XWA9EjGFGASBVeBiOBhCIuDB-P0CXryw2h5fpjNTqGHaoXl Go Links SnagIt
Launch Documents > Document Manager > Search: All Sources Log Out
Welcome Document Manager
Document Manager
Publish Save as Private Draft Cancel
Open File
Open the file for editing
Publish Save as Private Draft Cancel
Home Administration Documents Messaging Site Map
Done Internet

Create a new document

Type a file name and click Open File to begin editing. When you have finished, save and close the file in the application. The document must then be published or saved as a draft in order to be stored in the document library.

*File name: Nicole_Hargrove.ort

Title: Nicole_Hargrove

Description:

Language: English

* Required

Figure 8-62 Selecting Open file in Rich Text editor

17.The Rich Text editor will open in a separate window, as shown in Figure 8-63.

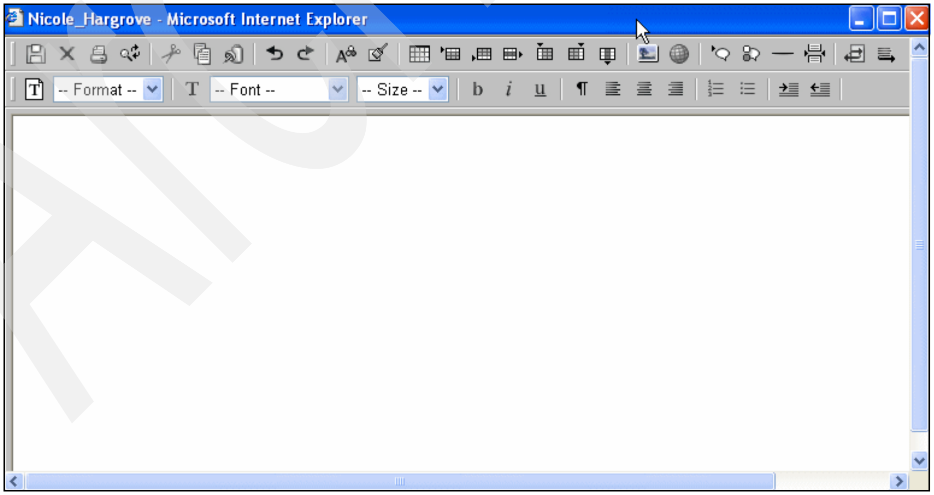
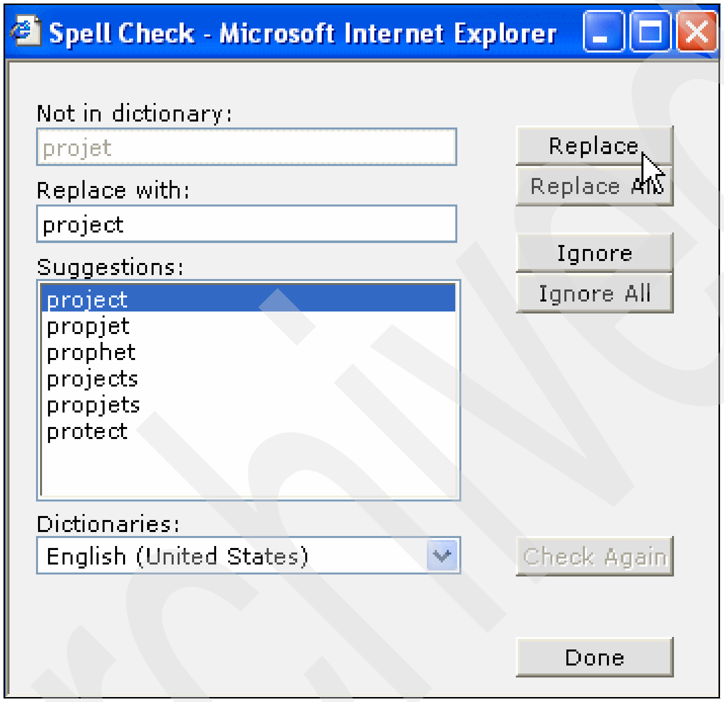


Figure 8-63 Rich Text editor

18. On the task bar, notice the Spell Checker icon abc (you may have to scroll to the right to see it).

19. Add some text. We have misspelled "project" and clicked to spell check the document.

20. The Spell Check opens in another window, as shown in Figure 8-64. Under Suggestions, select **project** and click **Replace**, as shown.



Spell Check - Microsoft Internet Explorer

Not in dictionary: projet

Replace with: project

Suggestions:

- project
- propjet
- prophet
- projects
- propjets
- protect

Dictionaries: English (United States)

Figure 8-64 Spell Check window

21. Click **Done** to close the Spell Check window and return to the Rich Text editor.

22. Click 💾 to save the document and then click ☒ to close the Rich Text editor.

23. Click **Publish** to make this document immediately visible.

9

Using Web Content Management with Portal

One of the key features of WebSphere Portal is Web Content Management (WCM). WCM manages Web content from the first draft to the final Web presentation.

In Chapter 8, “Configuring Web Content Management” on page 303, we introduce WCM and discuss the configuration of WCM. In the current chapter, we explain how to use WCM. However, you must configure WCM *before* proceeding with the current chapter. To complete this task, follow the steps provided in Chapter 8.

This book uses some of the recommendations found in the *Web Content Management Best Practices Guide*. You can find this guide at the following site:

http://www-128.ibm.com/developerworks/WebSphere/library/techarticles/0701_devos/0701_devos.html

Note: In this book we employ a simple WCM scenario that used content elements which were based on our example project. This test scenario is *not* provided in the Additional materials for this book.

To create your own WCM scenario, use content elements which are appropriate to your own enviornment.

IBM provides more elaborate WCM examples, such as the integrated content example and WCM JumpStart™.

The integrated content example uses not only WCM but also Personalization and Portal Document features to create Web content.

WCM JumpStart contains Intranet and extranet example Web sites and list portlets to make it easy to organize and access announcements, news, events, FAQs, and links. Both of these examples can be found in the Portal infocenter in an article entitled “Managing Web content”, which is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wcm/wcm_intro.html

9.1 Creating a new Web content library

We recommend that you create a new Web content library, because you cannot syndicate to an existing library with the same name.

Follow these steps to create a new Web content library called itso_wcm_library.

1. Login to the Portal as a WCM admin user.
2. Click **Launch -> Administration** to access the Portal administration portlets.
3. Click **Portal Content -> Web Content Libraries** as shown in Figure 9-1.

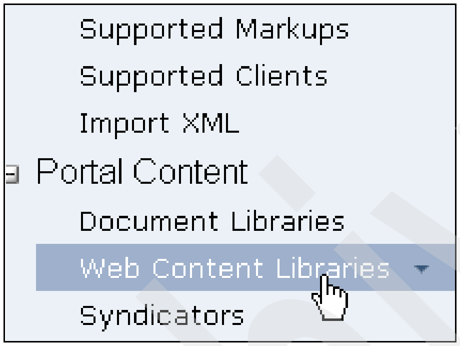
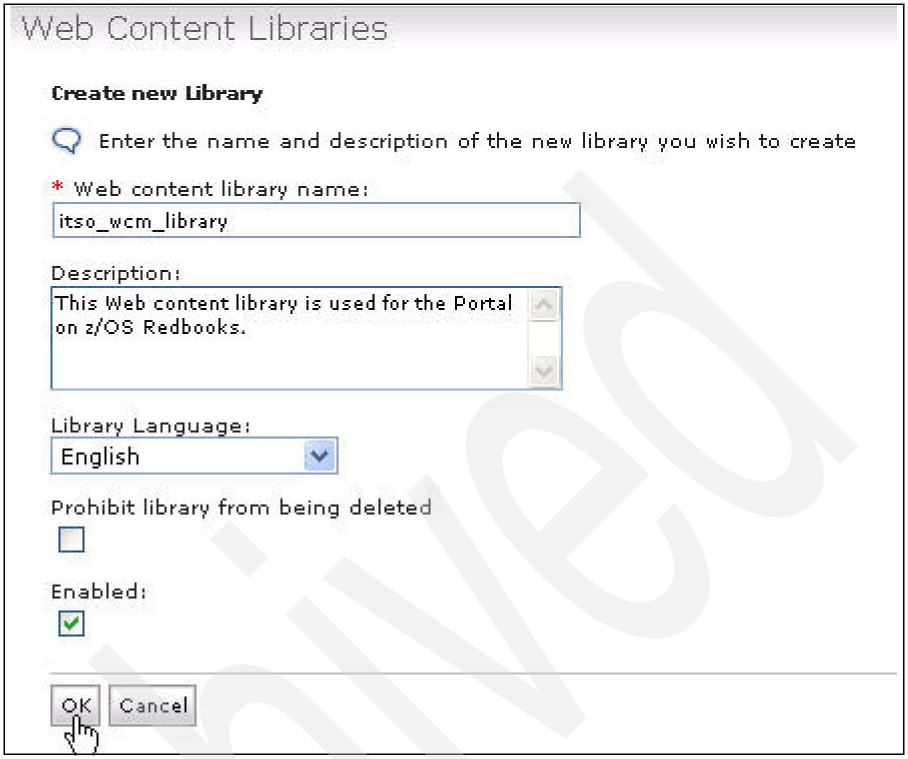


Figure 9-1 Selecting Web content libraries

4. Click Create new Library in the Web Content Libraries portlet.
5. Provide a Web content library name (required), description (optional), and click **OK** as shown in Figure 9-2 on page 386.

> **Note:** The language can only be set when the library is created, and it cannot be changed after creation.



Web Content Libraries

Create new Library

Enter the name and description of the new library you wish to create

* Web content library name: itso_wcm_library

Description: This Web content library is used for the Portal on z/OS Redbooks.

Library Language: English

Prohibit library from being deleted

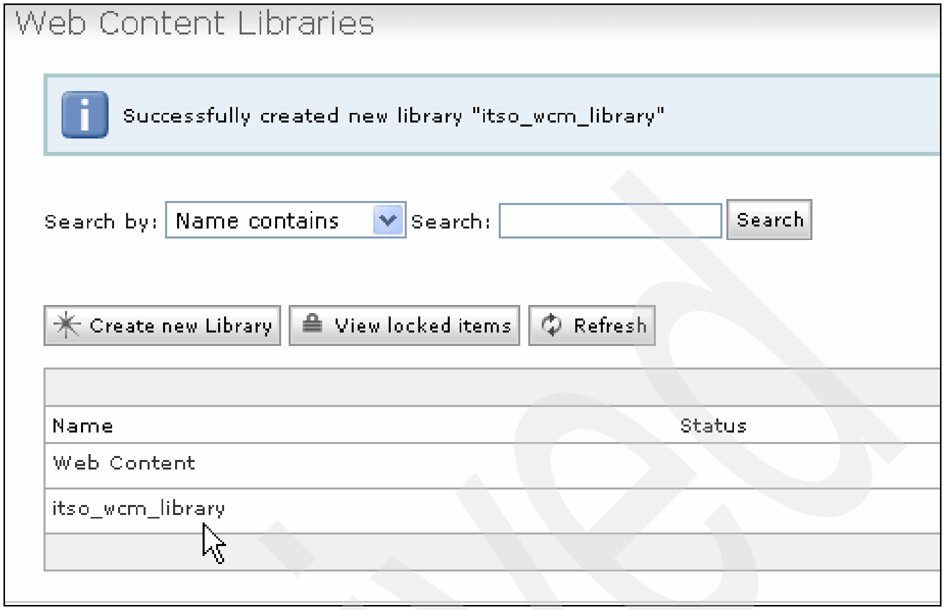
□

Enabled:

☑

Figure 9-2 New Web Content Library properties

6. After a few minutes the message `Successfully created new library` "`itso_wcm_library`" appears, and the newly created library will be in the list as shown in Figure 9-3 on page 387.



Successfully created new library "itso_wcm_library"

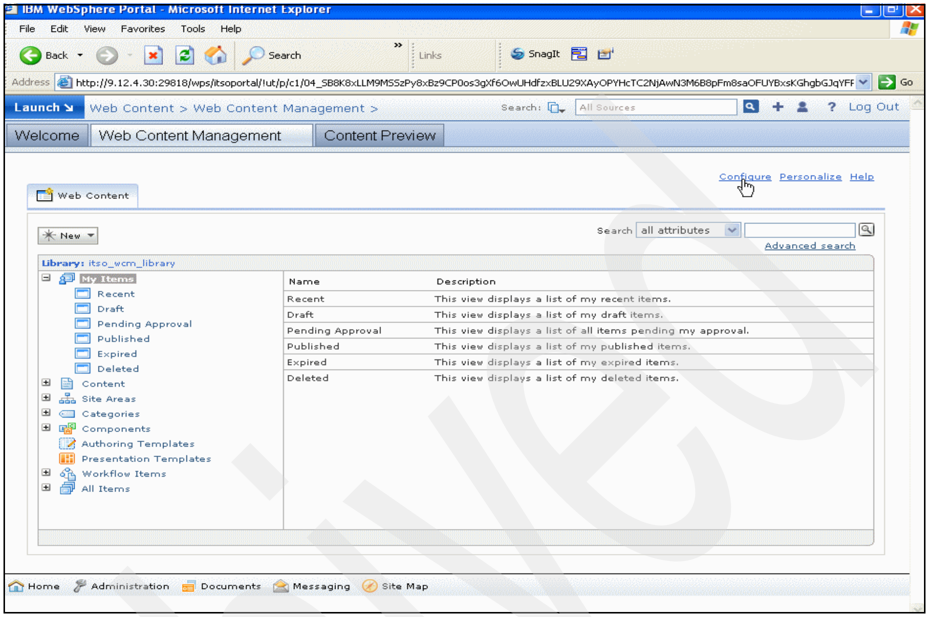
Search by: Name contains

Name	Status
Web Content	
itso_wcm_library	

Figure 9-3 Successfully created itso_wcm_library

Now you need to configure the WCM Authoring portlet to point to the itso_wcm_library. This will ensure that newly created items will be stored in the itso_wcm_library and not in the default library called Web Content.

7. Select **Configure** in the upper right corner of the WCM authoring portlet as shown in Figure 9-4.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Links
SnagIt
Address
http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3gXf6OwUHdfzxBLU29XAyOPYHcTC2NjAwN3M6B8pFm8saOFUYBxsKGhgbGJqYFF
Go
Launch
Web Content > Web Content Management >
Search:
All Sources
Log Out
Welcome
Web Content Management
Content Preview
Configure
Personalize
Help
Web Content
New
Search
all attributes
Advanced search
Home
Administration
Documents
Messaging
Site Map

Library: itso_wcm_library

- My Items
  - Recent
  - Draft
  - Pending Approval
  - Published
  - Expired
  - Deleted
- Content
- Site Areas
- Categories
- Components
- Authoring Templates
- Presentation Templates
- Workflow Items
- All Items

Name	Description
Recent	This view displays a list of my recent items.
Draft	This view displays a list of my draft items.
Pending Approval	This view displays a list of all items pending my approval.
Published	This view displays a list of my published items.
Expired	This view displays a list of my expired items.
Deleted	This view displays a list of my deleted items.

Figure 9-4 Selecting Configure to change libraries

8. Expand Library Selection by clicking ⊞.

   The box on the left displays the existing libraries. The box on the right lists the library that the Authoring portlet is currently configured to use.

9. Select **itso_wcm_library** from the list and click **Add** to move itso_wcm_library to the box on the right. Remove any other libraries that are currently selected, as shown in Figure 9-5 on page 389.

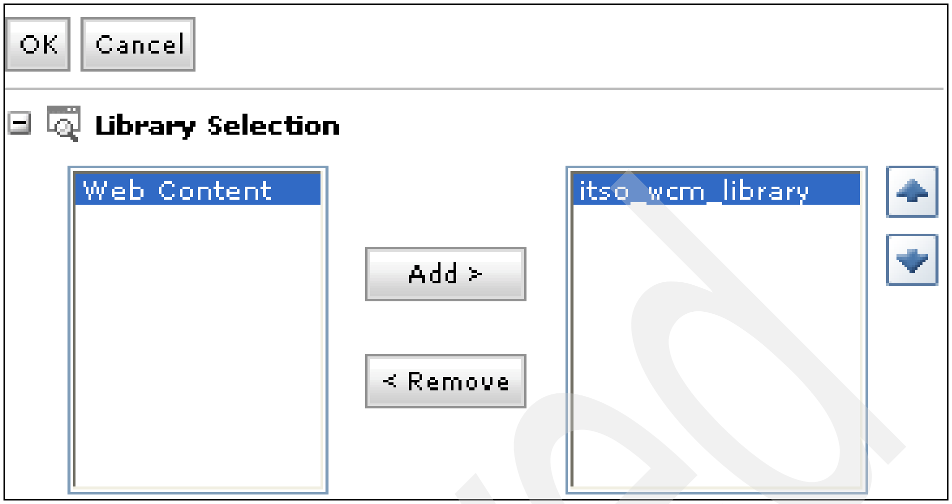


Figure 9-5 Designating itso_wcm_library as the library in use

10.Click **OK** to exit Configure.

11.Now the Library name should be itso_wcm_library as shown in Figure 9-6.

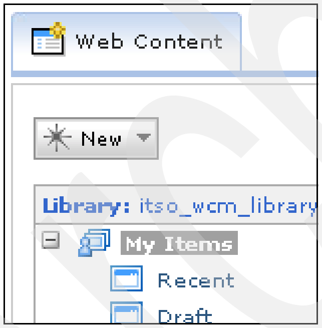


Figure 9-6 Working with the itso_wcm_library

9.2 Creating a workflow

WCM uses *workflows* to control the access to, verification and eventual approval of WCM items. An item must be approved at all stages up to a published stage before it can be viewed; this is typically called a *linear workflow*.

If an item is rejected at any stage it is sent back to the first stage of the workflow, regardless of whatever stage it is in during the approval process.

WCM workflows must have at least one stage. WCM only supports linear workflows. A reject stage can be specified, and comments can be added to items in each stage. These comments are added to the item’s history section.

Each workflow stage contains a set of actions, as shown in Table 9-1, that are executed when entering and exiting the stage.

Table 9-1 Workflow stage actions

Action	Description
Publish	Changes an item’s status from draft to publish to that it is available on the rendered site.
Expire	Changes an item’s status from published to expired making it no longer available on the rendered site.
E-mail	This sends e-mails when executed. A link to an item to be reviewed is included in the e-mail.
Scheduled Move	Performs a scheduled move to the next stage on a specified date.

Developing a workflow strategy is an important part of planning a WCM implementation. The Portal Infocenter provides more information about this topic in an article entitled “Developing a workflow strategy”, which is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wcm/wcm_dev_workflows.html

In the following steps you create a workflow that contains only a publish stage and publish action. Any item that uses this workflow will be available to view on the site.

9.2.1 Creating a workflow action

Follow these steps to create a workflow action called Publish.

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.

3. Click **New -> Workflow Actions -> Publish Action** as shown in Figure 9-7.

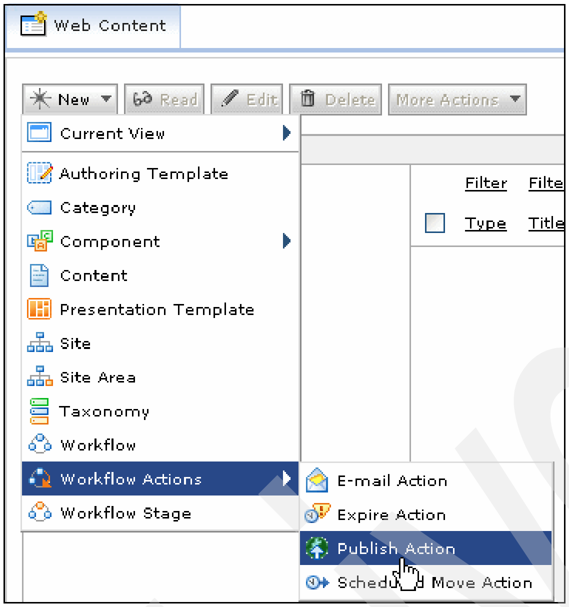


Figure 9-7 Selecting publish action

The workflow action form has three sections: Identification, Access, and History.

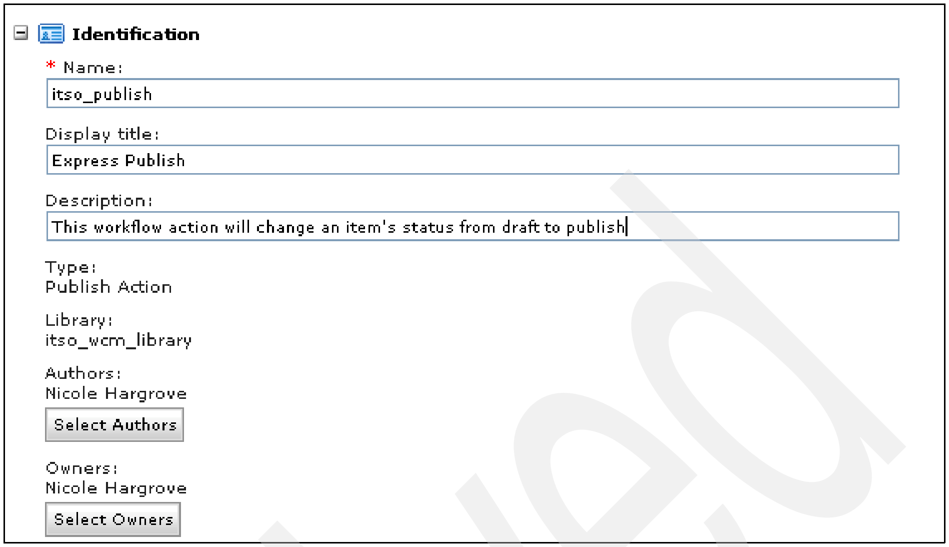
The Identification section captures action name, action display title, action description, item type, library where this action is stored, action authors, and action owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte or non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The Access section captures which users have access to the action and their level of access.

The History section captures comments that have been input during a workflow stage.

4. Specify the Name (required), Display title (optional), and Description (optional) for the publish action as shown in Figure 9-8 on page 392.



* Name:
itso_publish

Display title:
Express Publish

Description:
This workflow action will change an item's status from draft to publish

Type:
Publish Action

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-8 Publish workflow action identification properties

5. Accept the defaults for all other values. Click **Save... -> Save and Close** to save the workflow action and return to the initial WCM authoring portlet page.

9.2.2 Creating a workflow stage

Follow these steps to create a workflow stage called itso_wcm_stage and designate the Express Publish action to execute when entering this stage.

1. Click **New** -> **Workflow stage** as shown in Figure 9-9 on page 393.

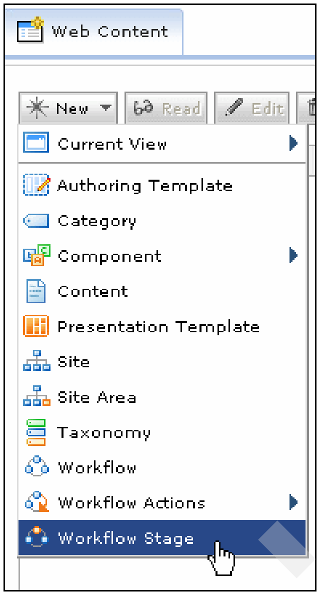


Figure 9-9 Selecting Workflow stage

The workflow stage form has four sections: Identification, Properties, Access, and History.

The Identification section captures stage name, stage display title, stage description, item type, library where this stage is stored, stage authors, and stage owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

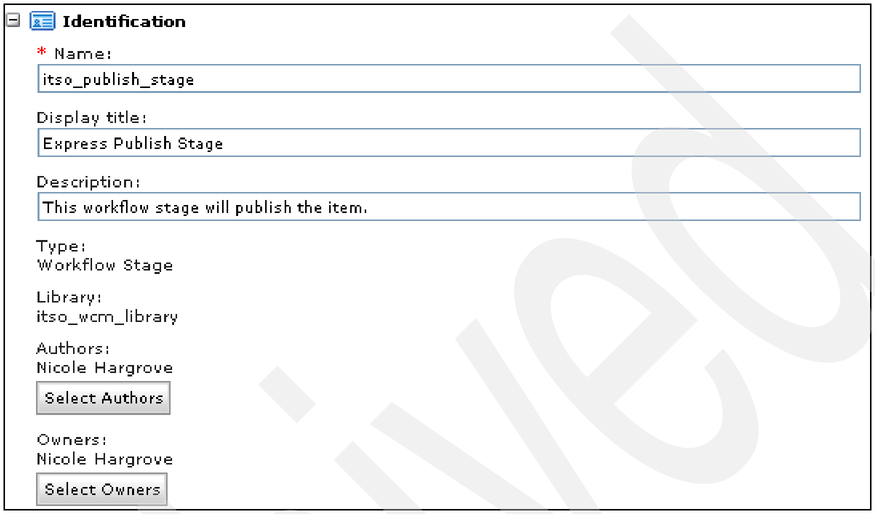
The Properties section includes which actions to execute when entering and exiting this stage, joint approval option, comments on approval option, and workflow security.

> **Note:** Joint approval requires every user, and at least one user from every group, granted “approval access” to approve the workflow stage.

The access section captures which users have access to an item and their level of access.

The History section captures comments that have been inputted during a workflow stage.

2. Specify the Name (required), Display title (optional), and Description (optional) for the Express Publish stage as shown in Figure 9-10.



Identification

* Name:
itso_publish_stage

Display title:
Express Publish Stage

Description:
This workflow stage will publish the item.

Type:
Workflow Stage

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-10 Express Publish workflow stage identification properties

3. Expand **Properties** and, under Execute on Entering Stage, click Select Actions.
4. Mark the checkbox to the left of Express Publish and click **OK** twice.
5. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the workflow stage and return to the initial WCM Authoring portlet page.

9.2.3 Creating a workflow

Follow these steps to create a workflow named itso_express_workflow with one stage - itso_publish_stage.

1. Click **New** -> **Workflow stage** as shown in Figure 9-11 on page 395.

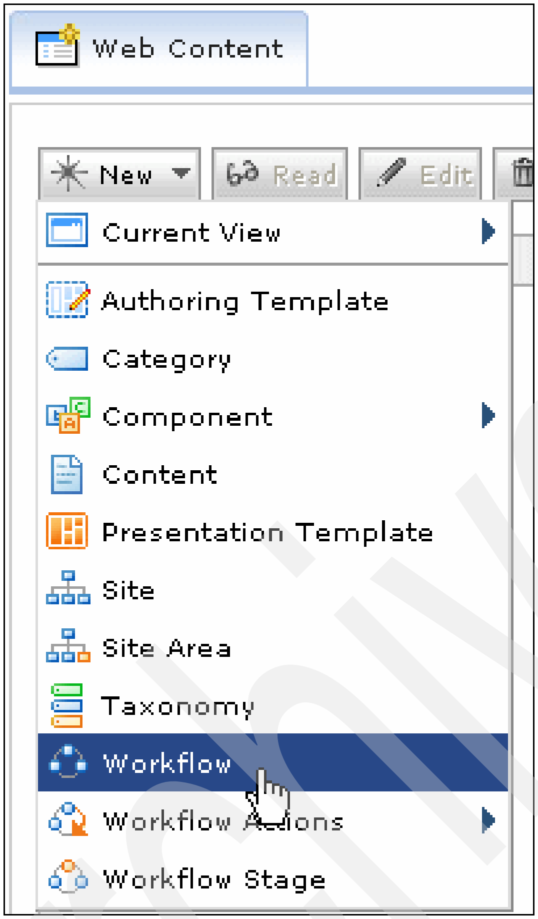


Figure 9-11 Selecting workflow

The workflow form has four sections: Identification, Properties, Access, and History.

The Identification section captures workflow name, display title, description, item type, library where this workflow stored, authors, and owners.

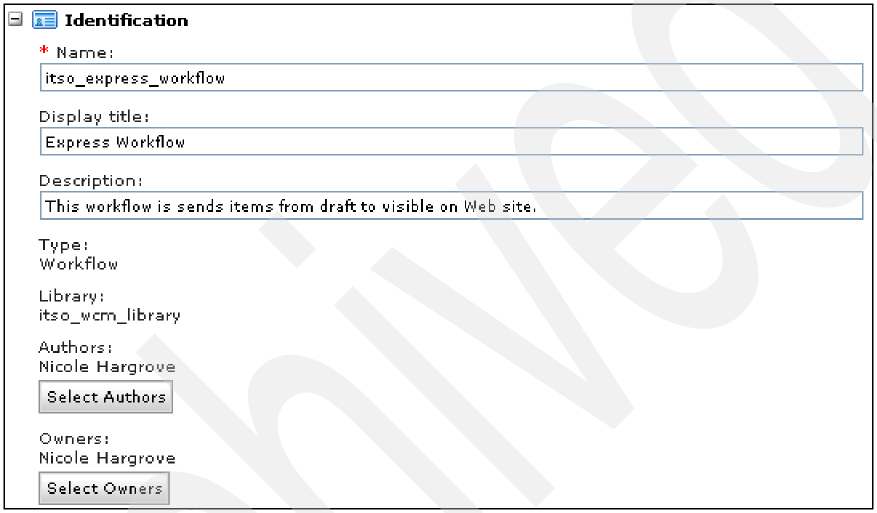
> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The Properties section includes workflow and reject stages, and comments on approval option.

The Access section captures which users have access to this workflow and their level of access.

The History section captures comments that have been inputted during a workflow stage.

2. Specify the Name (required), Display title (optional) and Description (optional) for the workflow as shown in Figure 9-12.



⊟ **Identification**

* Name:
itso_express_workflow

Display title:
Express Workflow

Description:
This workflow is sends items from draft to visible on Web site.

Type:
Workflow

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-12 itso_express_workflow identification properties

3. Expand **Properties** and under the Workflow stages click Select Workflow Stages.
4. Mark the checkbox to the left of Express Publish Stage and click **OK** twice to return to the Workflow form.
5. Accept the defaults for all other values and click **Save...-> Save and Close** to save the workflow and return to the initial WCM Authoring portlet.

9.3 Creating site framework

A *site framework* consists of site, site areas, and content items. The following steps take you through an example of creating a site and one site area.

9.3.1 Creating a site

Follow these steps to create a site called ITSO Site.

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.
3. Click **New** -> **Site** as shown in Figure 9-13.

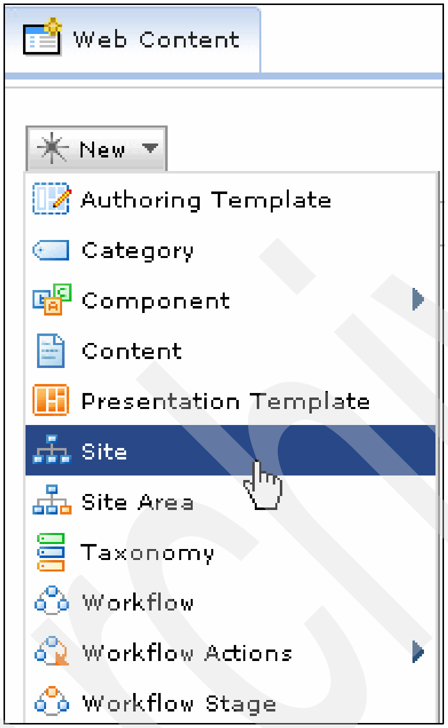


Figure 9-13 Selecting a new site

The site form has five sections: Identification, Search, Site Properties, Access, and History.

The Identification section captures site name, site display title, site description, item type, library where this site is stored, site authors, and site owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The Search section captures whether this item is searchable.

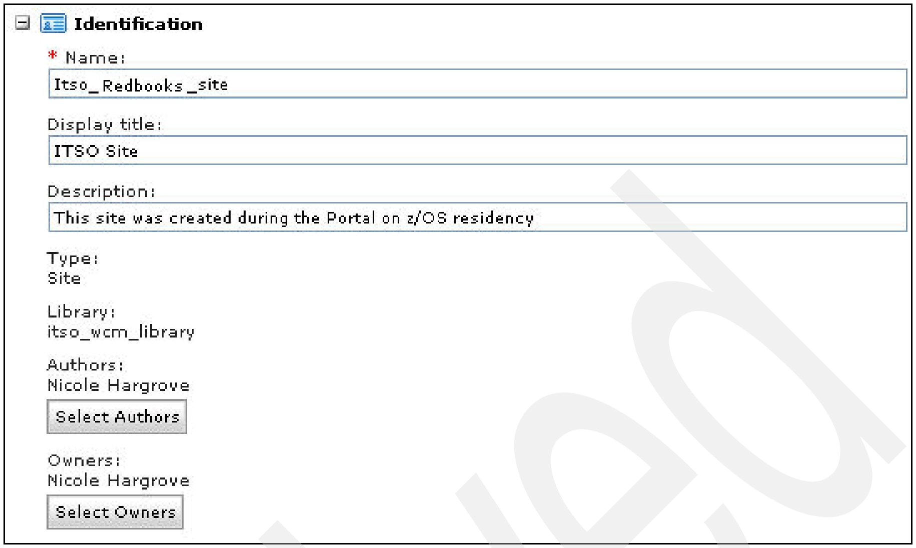
> **Note:** An administrator must create a search collection for the site before it can be made searchable.
>
> If the site author does not have the authority to create search indexes, an administrator will need to edit this site form and enter this information.

The Site Properties section gives you the option to pre-populate the site with content. You can create a site with blank content if no content items exist. This is also where the authoring and presentation templates are mapped to a site.

The Access section captures which users have access to the site and their level of access.

The History section captures comments that have been inputted during a workflow stage.

4. Specify the Name (required), Display title (optional), and Description (optional) for the workflow as shown in Figure 9-14 on page 399.



Identification

* Name:

Itso_Redbooks_site

Display title:

ITSO Site

Description:

This site was created during the Portal on z/OS residency

Type:
Site

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-14 Site identification properties

5. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the site and return to the initial WCM Authoring portlet page.

Next you will create a site area and associate it with the ITSO Site.

9.3.2 Creating a site area

Follow these steps to create a site area called ITSO Site Area. The ITSO Site Area is where your Web content will be displayed.

1. Click **New** -> **Site Area** as shown in Figure 2 on page 400.

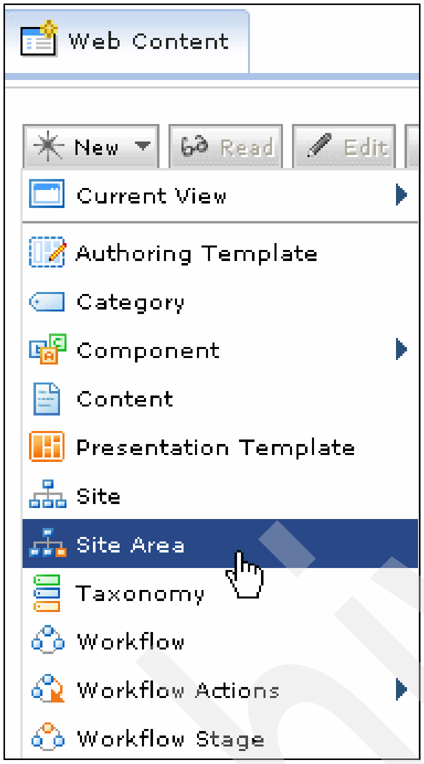


Figure 9-15 Selecting a new site area

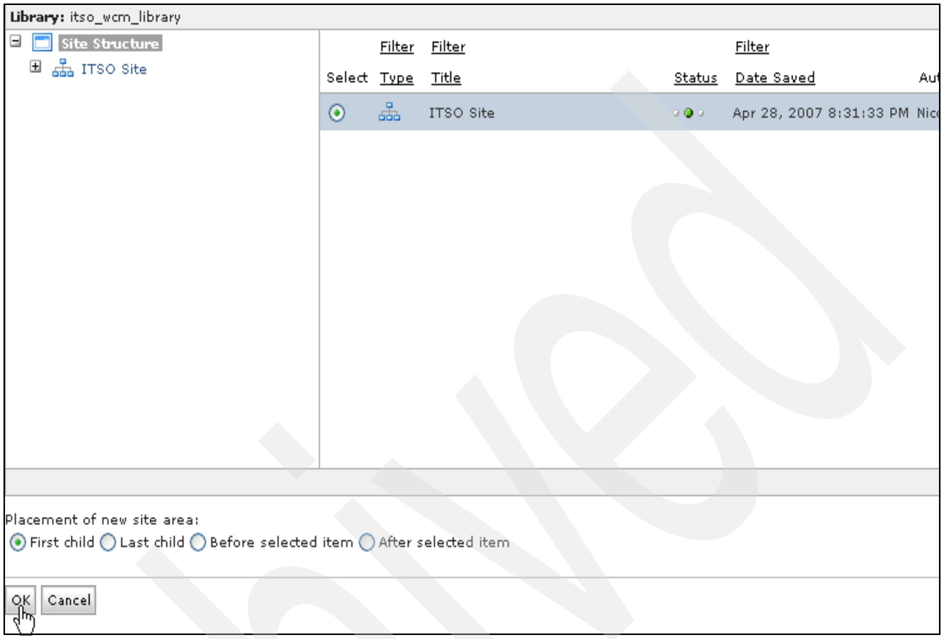
2. Select the ITSO Site radio button on the right.

   There are four options where a site area can be placed: first child, last child, before the selected item, or after the selected item.

   If the first or last child is selected, the new site is placed as the first or last child of the site or site area. If before or after the selected item is selected, the new site is placed before or after the select item (in this case, a site area).

3. Select **First child** under Placement of the new site area (because you only have one site area).

4. Your form should resemble Figure 9-16. Now click **OK**, as shown in Figure 9-6 on page 389.



Library: itso_wcm_library

Site Structure
ITSO Site

	Filter	Filter		Filter	
Select	Type	Title	Status	Date Saved	Aut
●		ITSO Site		Apr 28, 2007 8:31:33 PM	Nic

Placement of new site area:
● First child ○ Last child ○ Before selected item ○ After selected item

Figure 9-16 Associating a site with the site area

Next you will see the site area form. The site area form has four sections: Identification, Site Area Properties, Access, and History.

The Identification section captures site area name, site area display title, site area description, item type, library where this site area is stored, site authors, and site owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Display Title field can have double-byte and non-ASCII characters.

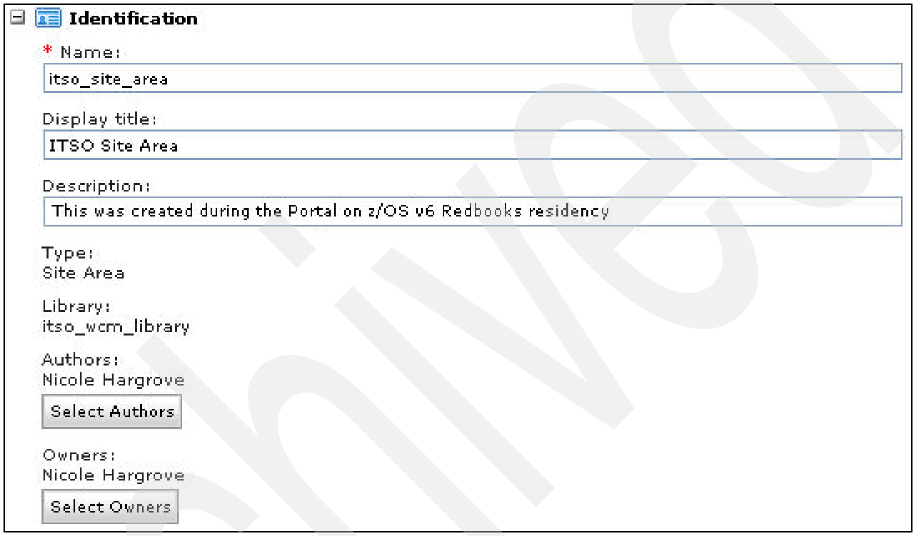
The Site Area Properties section includes default content to be displayed on the site, workflow, and the template map. The workflow section is only visible if the workflow feature has been enabled for site areas. Template maps create links between authoring templates and presentation templates.

Template maps determine which presentation template will be used to display the elements stored in a content item.

The Access section captures which users have access to this site area and their level of access.

The History section captures comments that have been inputted during a workflow stage.

5. Specify the Name (required), Display title (optional) and Description (optional) for the site area as shown in Figure 9-17.



⊟ **Identification**

* Name:
itso_site_area

Display title:
ITSO Site Area

Description:
This was created during the Portal on z/OS v6 Redbooks residency

Type:
Site Area

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-17 Site Area identification properties

6. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the site area and return to the initial WCM Authoring portlet page.

Now you can create an authoring template for the home page of the ITSO Site.

9.4 Creating an authoring template

Follow these steps to create an authoring template named itso_home_page. Any new content that gets created for pages will use this authoring template.

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.

3. Click **New** -> **Authoring Template** as shown in Figure 9-18.

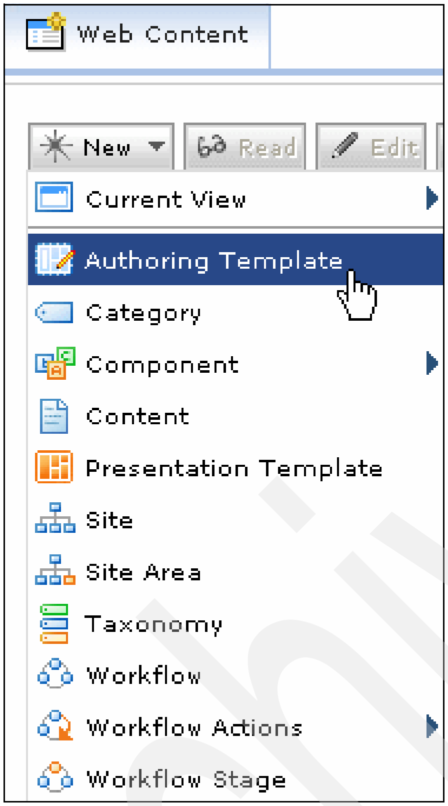


Figure 9-18 Selecting New -> Authoring Template

Next you will see the authoring template form. The authoring template form has four sections: Identification, Content Form Properties and History.

The Identification section captures authoring template name, authoring template display title, authoring template description, item type, library where this authoring template is stored, authoring template authors, and authoring template owners.

Note: The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.

Do not use double-byte nor non-ASCII characters for the Name field only.

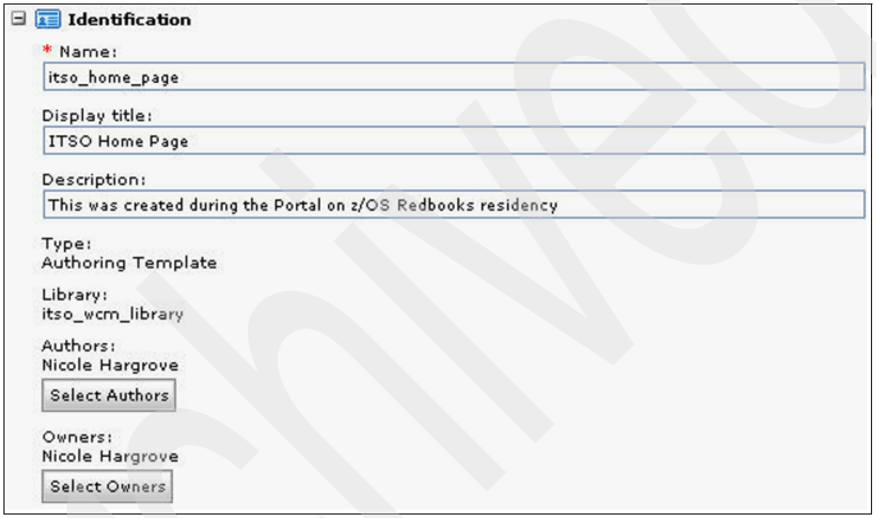
The Title field can have double-byte and non-ASCII characters.

The Content Form Properties section specifies the characteristics of the content form generated from the authoring template. This can include how sections are displayed, the sites and site areas where content can be saved, and customized help text for the content form.

The Access section captures which users have access to the authoring template and their level of access.

The History section captures comments that have been inputted during a workflow stage.

4. Specify the Name (required), Display title (optional) and description (optional) for the authoring template as shown in Figure 9-19.



⊟ **Identification**

* Name:
itso_home_page

Display title:
ITSO Home Page

Description:
This was created during the Portal on z/OS Redbooks residency

Type:
Authoring Template

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-19 ITSO home page authoring template

5. Click **OK** to return to the Authoring Template form.

   Now designate the Express workflow as the default workflow to all content created using this authoring template.

6. Expand the workflow section and click **Select Workflow**.
7. You will see a list of workflows at have been created. In this case, you only have one workflow so select the radio button next to Express workflow and click **OK**.
8. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the authoring template and return to the initial WCM Authoring portlet page.

9.4.1 Creating Web content with the authoring template

Now you need to create a new piece of content called itso_hp_content using the authoring template to be designated as content for the ITSO Site.

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.
3. Click **New** -> **Content** as shown in Figure 9-20.

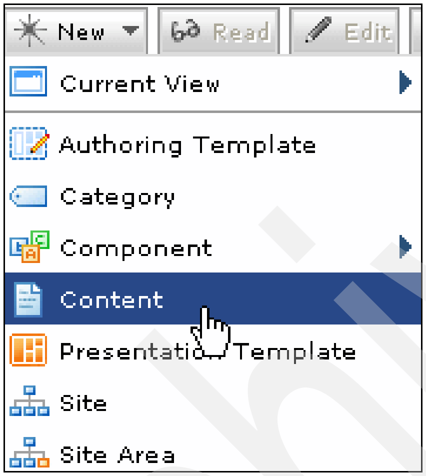


Figure 9-20 Selecting new content

You will see a list of authoring templates. In this case you only have one listed because you have only created one.

4. Select the ITSO Home Page authoring template you previously created and click **OK**.

You will see the content form. The content form has six sections: Identification, Profile, Content Properties, Workflow, Access and History.

The Identification section captures content name, content display title, content description, item type, library where this content is stored, content authors, and content owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The Profile section captures personalized Web content. In a profile-based personalized site, although users may not be able to access all the pages via personalized menus, they may still be able to access other pages by using navigators, or by searching for content. Using access controls limits users to only view items that they have been granted access to.

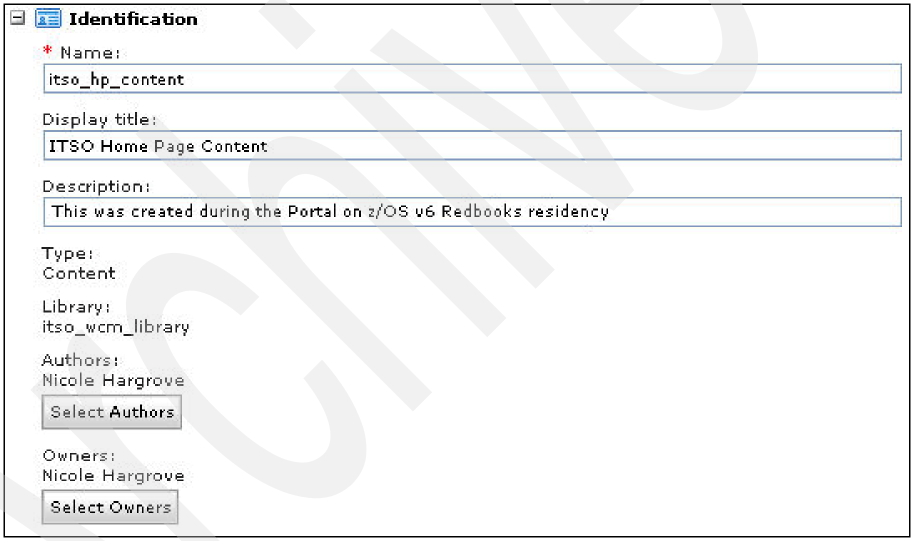
The Content Properties section specifies the authoring template in use.

The Workflow section specifies document status, current workflow and workflow stage, next workflow stage, publish and expiry date, and actions.

The Access section captures which users have access to the content and their level of access.

The History section captures comments that have been inputted during a workflow stage.

5. Specify the Name (required), Display title (optional), and Description (optional) in the identification section as shown in Figure 9-21.



⊟ **Identification**

* Name:
itso_hp_content

Display title:
ITSO Home Page Content

Description:
This was created during the Portal on z/OS v6 Redbooks residency

Type:
Content

Library:
itso_wcm_library

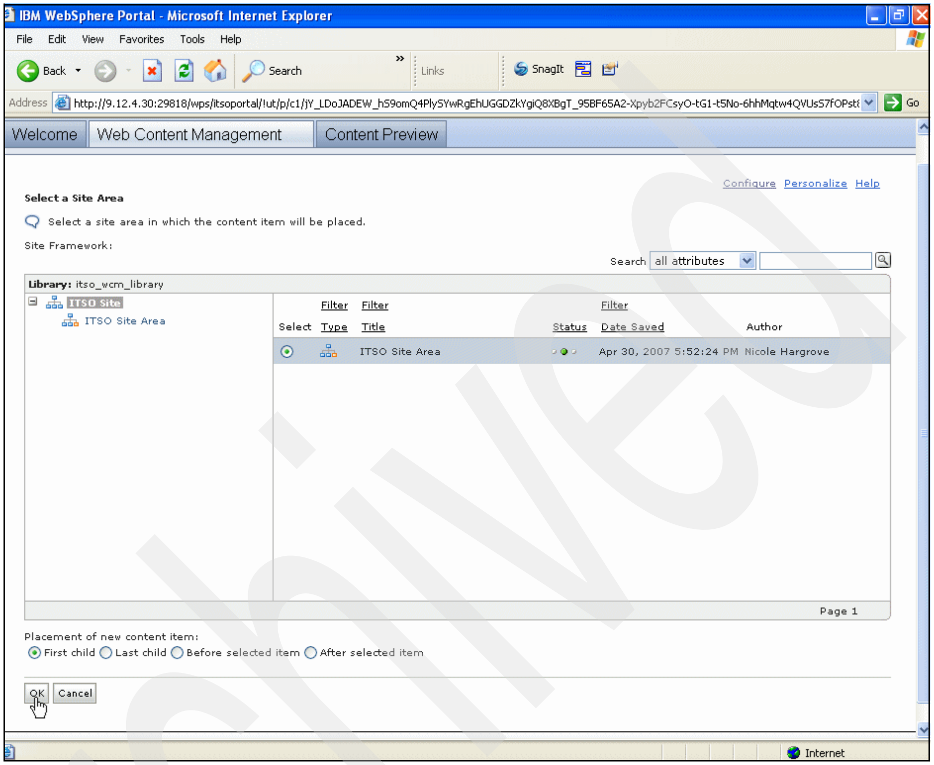
Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-21 itso_hp_content identification properties

6. Click **Save**. The next page allows you to select where the content will be displayed on the site.

7. Select **ITSO Site** on the left side of the screen, select the **ITSO Site Area** radio button, select the **First Child** radio button, and then click **OK** as shown in Figure 9-22.



http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jY_LDoJADEW_hS9omQ4PlySYwRgEhUGGDZkYgiQ8XBgT_95BF65A2-Xpyb2FCsyO-tG1-t5No-6hhMqtw4QVUsS7fOPst(

Select a Site Area

Select a site area in which the content item will be placed.

Site Framework:

Search all attributes

Library: itso_wcm_library

ITSO Site
ITSO Site Area

Select	Filter Type	Filter Title	Status	Filter Date Saved	Author
●		ITSO Site Area		Apr 30, 2007 5:52:24 PM	Nicole Hargrove

Page 1

Placement of new content item:
● First child ○ Last child ○ Before selected item ○ After selected item

OK Cancel

Figure 9-22 Selecting where to display new content

8. You will be returned to the initial WCM Authoring portlet page.

Now you will create the content that will be referenced in the presentation template.

9.5 Creating ITSO content

This section explains how to create content to view in portlets and Web sites. This content will be referenced later in the presentation template. Table 9-2 on page 408 lists the six image components and their names.

Table 9-2 Image components

Image component	Image name
IBM Redbooks Logo	Redbooks_logo
Additional IBM Redbooks Teammember - Doris Fiorentino	art_dorisf
Original IBM Redbooks Team	itso_ort
ITSO Systems Programmer Guru - Richard Conway	itso_e_rich
ITSO Project Leaders - G. Michael Connolly	itso_rpl_gmc
ITSO Project Leaders - Alex Louwe-Kooijmans	itso_rpl_alk

Now we create an HTML component. Table 9-3 lists the HTML component and its name.

Table 9-3 HTML component

HTML component	name
ITSO Portal on z/OS Redbooks players	itso_html_team

1. Click **Launch ->Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.
3. Click **New -> Component -> Image** as shown in Figure 9-23.

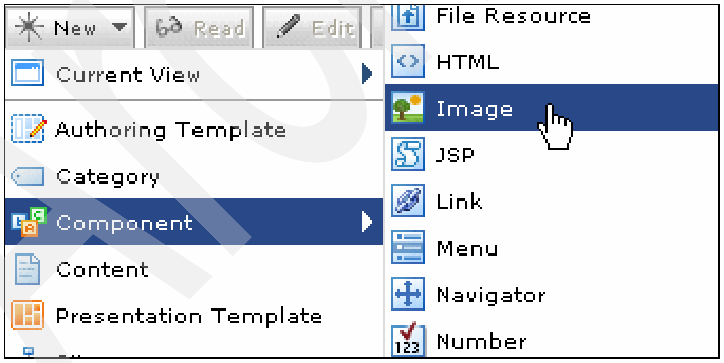


Figure 9-23 Selecting new image component

Next you will see the image component form. The image component form has four sections: Identification,Image Element, Access, and History.

The Identification section captures image component name, image component display title, image component description, item type, library where this image component is stored, image component authors, and image component owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

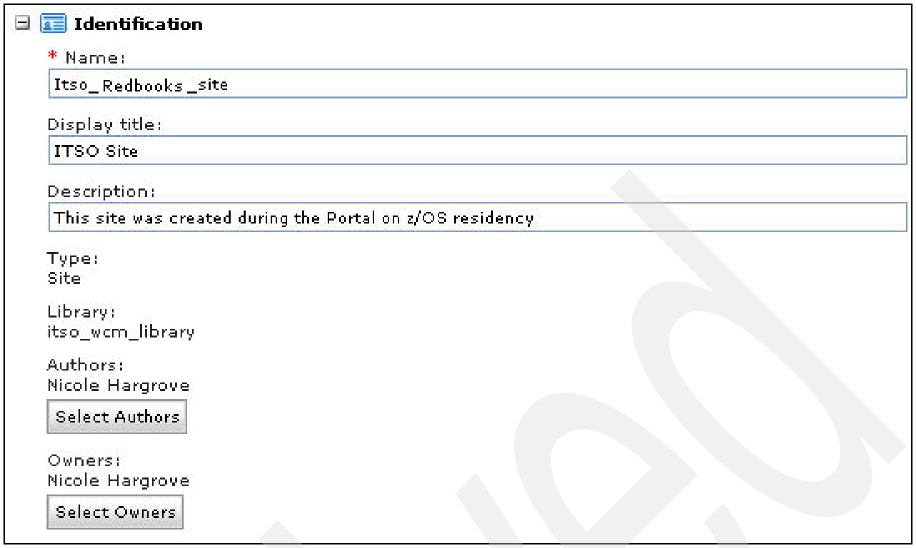
The Content Form Properties section specifies the characteristics of the content form generated from the authoring template. This can include how sections are displayed, the sites and site areas where content can be saved, and customized help text for the content form.

The Access section captures which users have access to the authoring template and their level of access.

The History section captures comments that have been input during a workflow stage.

> **Note:** By default, 10 MB is the largest single file that you can import. If you need to import files larger than 10 MB, an administrator will need to configure your server's inbound buffer size.
>
> For more information about this topic, refer to “Increasing the inbound buffer size”, which is available at the following site:
>
> `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wcm/wcm_config_importbuffer.html`

4. Specify the Name (required), Display title (optional), Description (optional) in the identification section as shown in Figure 9-24 on page 410.



⊟ **Identification**

* Name:
Itso_Redbooks_site

Display title:
ITSO Site

Description:
This site was created during the Portal on z/OS residency

Type:
Site

Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-24 Image component identification properties

5. Expand the Image Element section and click **Browse**.
6. Select your image form and click **Open**.
7. Provide values for Alternative Text and HTML Tag name.

8. Click **Save**; your screen should resemble Figure 9-25

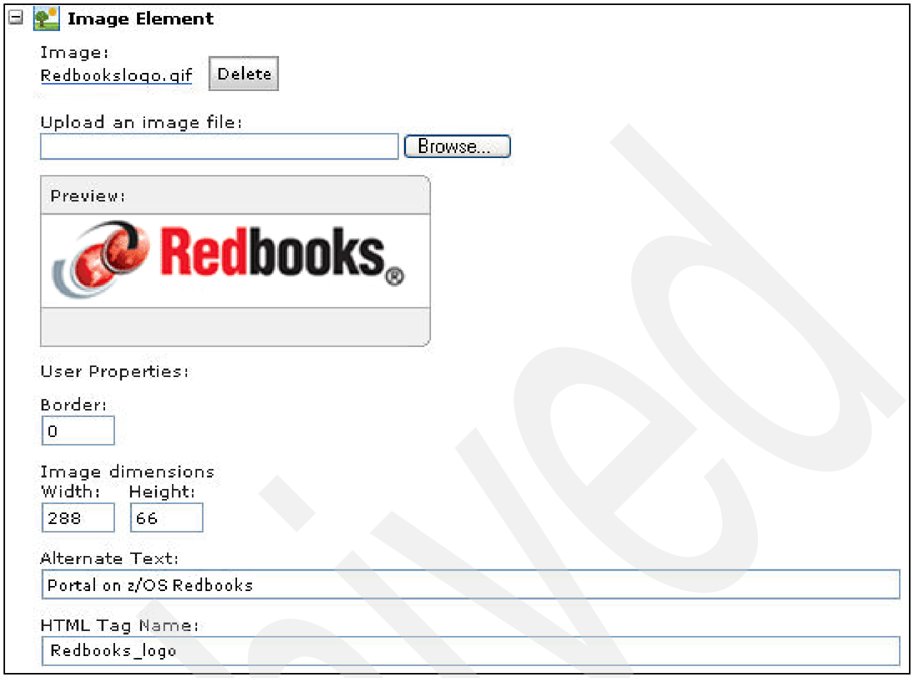


Image Element

Image:
Redbookslogo.gif

Upload an image file:

Preview:

User Properties:

Border:
0

Image dimensions
Width: 288
Height: 66

Alternate Text:
Portal on z/OS Redbooks

HTML Tag Name:
Redbooks_logo

Figure 9-25   Image component image element section

9. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the image component and return to the initial WCM Authoring portlet page.
10. Use the previous steps to create the other five image components.

Now you will create the HTML component:

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be present with the WCM Authoring portlet.

3. Click **New** -> **Component** -> **HTML** as shown in Figure 9-26.

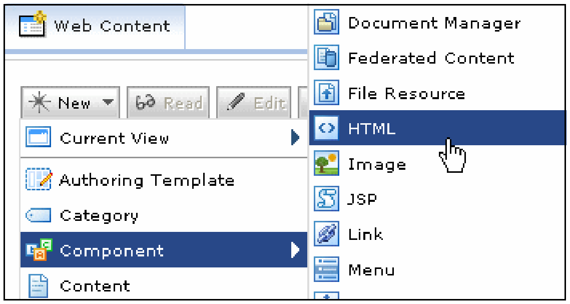


Figure 9-26 Creating a new HTML component

Next you will see the HTML component form. The HTML component form has four sections: Identification, HTML Element, Access and History.

The Identification section captures HTML component name, HTM component display title, HTM component description, item type, library where this HTM component is stored, HTM component authors, and HTM component owners.

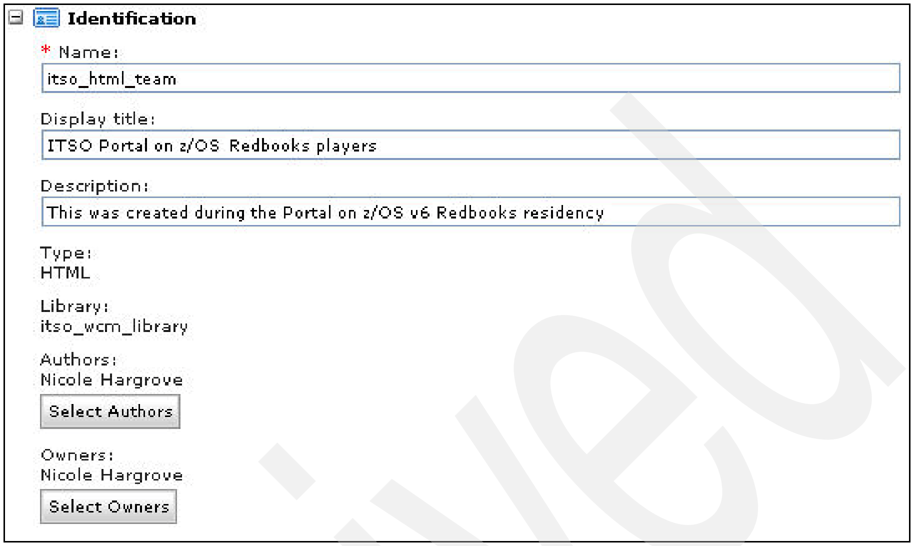
> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The HTML Element section is where you can input HTML tags.

The Access section captures which users have access to the HTML component and their level of access.

The History section captures comments that have been input during a workflow stage.

4. Specify the Name (required), Display title (optional), Description (optional) in the Identification section as shown in Figure 9-27.



* Name:
itso_html_team

Display title:
ITSO Portal on z/OS Redbooks players

Description:
This was created during the Portal on z/OS v6 Redbooks residency

Type:
HTML

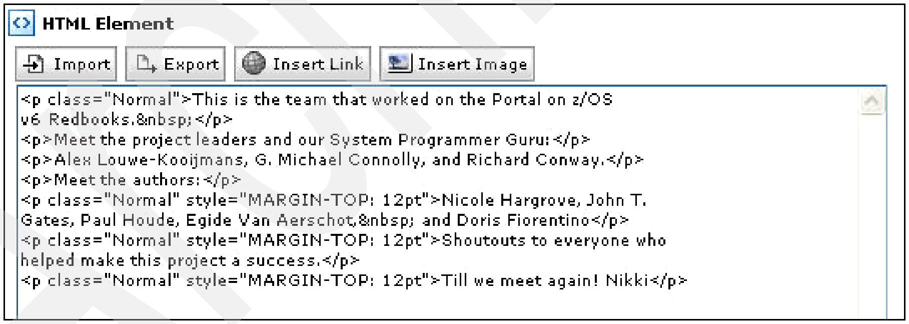
Library:
itso_wcm_library

Authors:
Nicole Hargrove

Owners:
Nicole Hargrove

Figure 9-27 HTML component identification properties

5. Copy the text from the html_element_snippet.txt into the HTML Element section as shown in Figure 9-28.



```
<p class="Normal">This is the team that worked on the Portal on z/OS
v6 Redbooks.&nbsp;</p>
<p>Meet the project leaders and our System Programmer Guru:</p>
<p>Alex Louwe-Kooijmans, G. Michael Connolly, and Richard Conway.</p>
<p>Meet the authors:</p>
<p class="Normal" style="MARGIN-TOP: 12pt">Nicole Hargrove, John T.
Gates, Paul Houde, Egide Van Aerschot,&nbsp; and Doris Fiorentino</p>
<p class="Normal" style="MARGIN-TOP: 12pt">Shoutouts to everyone who
helped make this project a success.</p>
<p class="Normal" style="MARGIN-TOP: 12pt">Till we meet again! Nikki</p>
```

Figure 9-28 HTML Element with code from snippet text file

This HTML code displays the text in the ITSO WCM CPV - Document portlet.

6. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the HTML component and return to the initial WCM Authoring portlet page.

Now that you have created the components, you need to create a presentation template that will reference the components.

9.6 Creating presentation templates

In the following steps you will create one presentation template called itso_home_page_pt. You will copy some preexisting code from the itso_home_page_pt_snippet.txt file. This HTML code will display the previously created HTML and image components in table format.

The ITSO Documents PT will be used to display the itso_html_team html component. The ITSO Leaders PT will be used to display the itso_enablement, itso_Redbooks_project_leaders_gmc, and itso_Redbooks_project_leaders_alk image components. The ITSO Authors PT will be used to display the itso_orig_Redbooks_team and Additional _Redbooks_Teammember components.

1. Click **Launch -> Web Content** to work with the WCM authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.
3. Click **New -> Presentation Template** as shown in Figure 9-29.

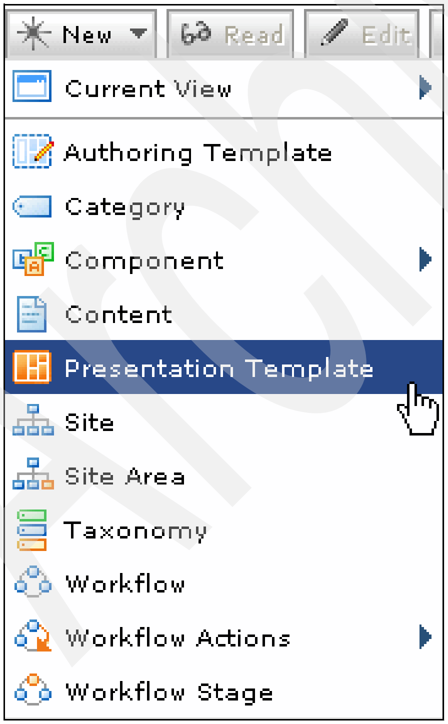


Figure 9-29 Creating a new presentation template

Next you will see the presentation template form. The presentation template form has four sections: Identification, HTML Element, Access, and History.

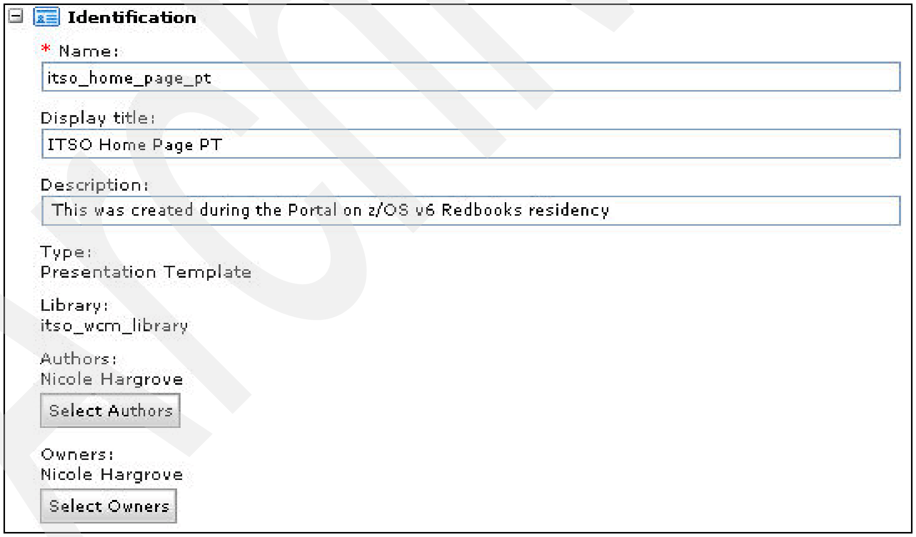
The Identification section captures presentation template name, presentation template display title, presentation template description, item type, library where this presentation template is stored, presentation template authors, and presentation template owners.

> **Note:** The Name field can contain only alphanumeric characters (a-z, A-Z, 0-9), the characters $ - _ . + ! ( ) and spaces.
>
> Do not use double-byte nor non-ASCII characters for the Name field only.
>
> The Title field can have double-byte and non-ASCII characters.

The Access section captures which users have access to the presentation template and their level of access.

The History section captures comments that have been inputted during a workflow stage.

4. Specify the Name (required), Display title (optional) and Description (optional) in the Identification section as shown in Figure 9-30.



⊟ **Identification**

* Name:
itso_home_page_pt

Display title:
ITSO Home Page PT

Description:
This was created during the Portal on z/OS v6 Redbooks residency

Type:
Presentation Template

Library:
itso_wcm_library

Authors:
Nicole Hargrove
Select Authors

Owners:
Nicole Hargrove
Select Owners

Figure 9-30 itso_home_page_pt identification properties

5. Delete any existing code in the Presentation Template section. Paste the code found in the itso_home_page_pt_snippet.txt into the Presentation Template section.

6. Accept the defaults for all other values and click **Save...** -> **Save and Close** to save the presentation template component and return to the initial WCM Authoring portlet page.

Now that you have all the pieces created for the site, you need to modify the ITSO site properties.

9.7 Designating the default content and template map

In the following steps you change the ITSO site properties to select the itso_hp_content as its default content and ITSO Home Page and ITSO Home Page PT as the authoring and presentation templates, respectively.

1. Click **Launch** -> **Web Content** to work with the WCM Authoring portlet.
2. Click the **Web Content Management** tab to be presented with the WCM Authoring portlet.
3. Expand **Site Areas** -> **By Site** on the left side, check **ITSO Site** on the right side, and click **Edit**.
4. Expand the **Site Properties** section and click **Select Default Content**.
5. Expand **ITSO Site** -> **ITSO Site Area**. Select the content item previously created (ITSO Home Page Content). Select **ITSO Site Area** on the left, then check the **ITSO Home Page Content** radio button and click **OK**.
6. You are returned to the Site form. Click **Edit Template Mapping** and click **Add** to add the template map.
7. On this page, under the Authoring Template section, check the **ITSO Home Page** radio button; under the Presentation Templates section, check **ITSO Home Page PT** radio button; then click **OK** twice.
8. You are returned to the Site form. Click **Save and Close** to return to the initial WCM Authoring portlet page.

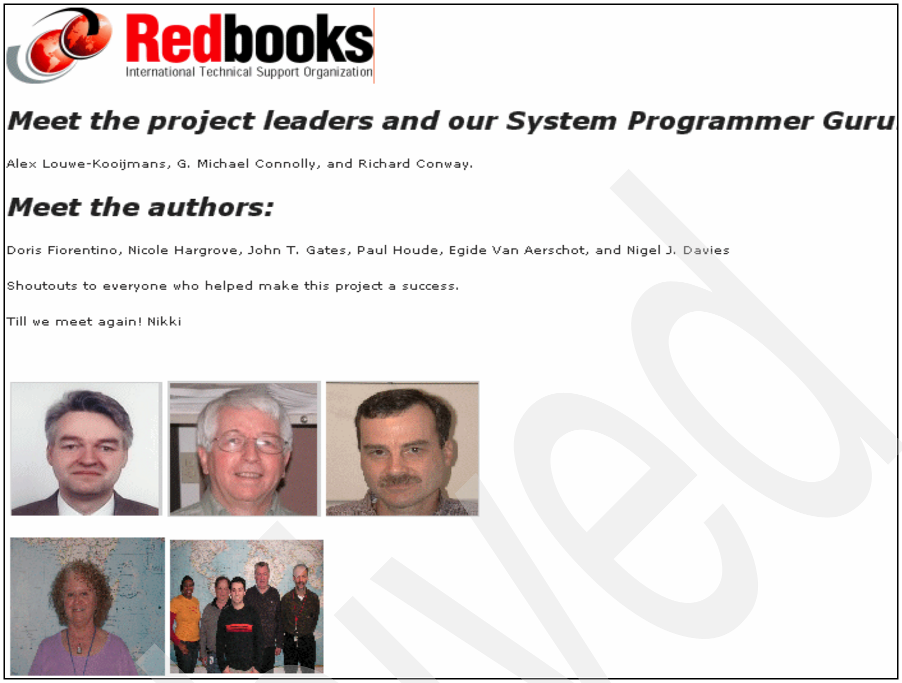
9.8 Configuring content viewer portlets

Now you will use the Content Preview portlet to view the content you just created. The Content Preview portlet gets deployed as a part of configuring the WCM Authoring portlet.

This portlet can be copied and configured to display different artifacts created with WCM.

In the following steps you configure the Content Viewer portlet to point to the ITSO Portal on z/OS Redbooks players component and ITSO Home Page content. Finally, you will change the Portlet Display Title to Portal on z/OS v6 Redbooks.

1. Click the **Content Preview** tab and click the **Portlet Menu** in the upper right corner of the portlet.
2. Click **Configure** and expand the **Content** section.
3. Click **Component** under the Content Type.
4. Click **Edit** under Component. Do not be alarmed if the previously created components are not in the list. You must change the library from Web Content to itso_wcm_library.
5. Select **itso_wcm_library** from the Library drop-down list.
6. Now you will see the components that you stored in the itso_wcm_library.
7. Select the **ITSO Portal on z/OS Redbooks players** radio button and click **OK**.
8. Click **Edit** under Content. Again do not be alarmed if the previously-created content is not listed. You must change the library from Web Content to itso_wcm_librar.
9. Select **itso_wcm_library** from the Library drop-down list.
10. Expand **Content by Site Area** -> **ITSO Site**.
11. Select **ITSO Site Area** on the left, and then select **ITSO Home Page Content** on the right and click **OK**.
12. To change the Portlet Display Title, expand **Settings** section and type `Portal on z/OS v6 Redbooks` in Field.
13. Click **OK**; then your screen should look similar to Figure 9-31 on page 418.



Redbooks

International Technical Support Organization

Meet the project leaders and our System Programmer Guru

Alex Louwe-Kooijmans, G. Michael Connolly, and Richard Conway.

Meet the authors:

Doris Fiorentino, Nicole Hargrove, John T. Gates, Paul Houde, Egide Van Aerschot, and Nigel J. Davies

Shoutouts to everyone who helped make this project a success.

Till we meet again! Nikki

Figure 9-31 Portal on z/OS v6 IBM Redbooks portlet

10

Integrating WebSphere Portal with host systems

Many WebSphere Portal on z/OS users will want to integrate their host applications and data in their WebSphere Portal environment. Although there are many ways to accomplish this integration, in this chapter we provide steps for implementing three solutions to accomplish this integration task.

This chapter contains the following three examples:

- Creating and deploying a Host On-Demand portlet
- Creating and deploying a CICS Web service client portlet application
- Configuring a MY Query Reports sample portlet

10.1 Creating and deploying a Host On-Demand portlet

Host On-Demand (HOD) provides the Deployment Wizard to create HOD applications. Instead of generating output files of HTML, you can generate output files of portlet. We used this tool to create a portlet that will display a TSO session.

Before you proceed with this section, we recommend that you become familiar with administering Portal. Refer to “Take a test drive of the site” in the Portal Infocenter to learn how to navigate Portal, create pages, and add portlets to a page. The article is available at the following site:

http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.docs/wps/gs.html

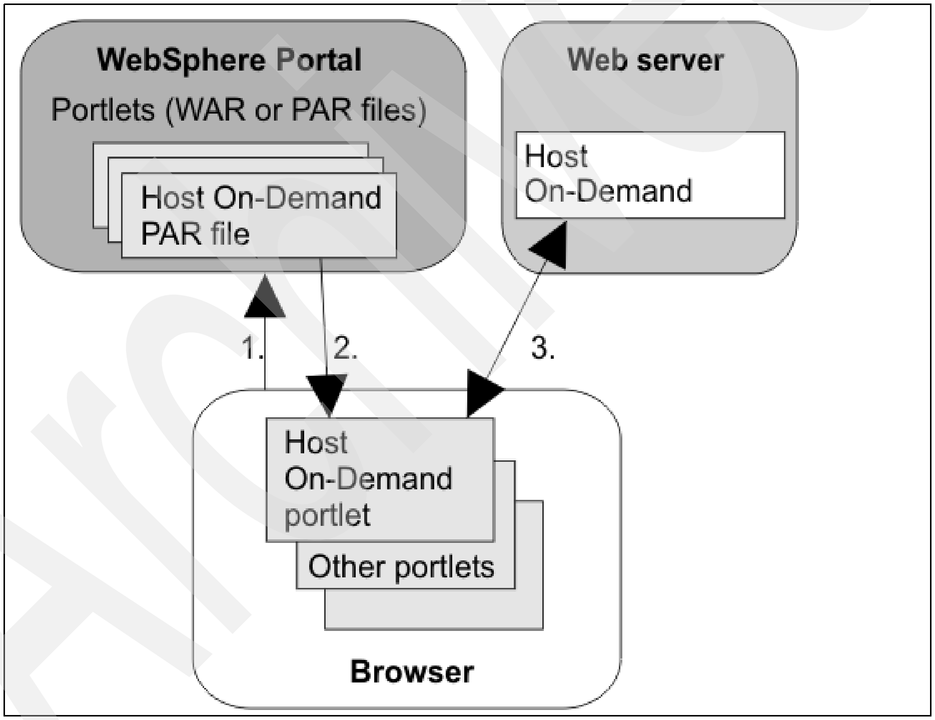


Figure 10-1 Portal and HOD application flow

Figure 10-1 depicts how HOD works with WebSphere Portal:

1. A user logs into the Portal via a Web browser and is authenticated by a user ID and password.

2. The user's customized set of portlets is downloaded to the user's machine and is displayed in the browser.
3. If the user has configured a Host On-Demand portlet, then Host On-Demand starts. This gives the user full Host On-Demand functionality within the portlet window, including being able to start sessions and perform other Host On-Demand tasks.

For details on how to install the HOD Deployment Wizard, refer to the following site:

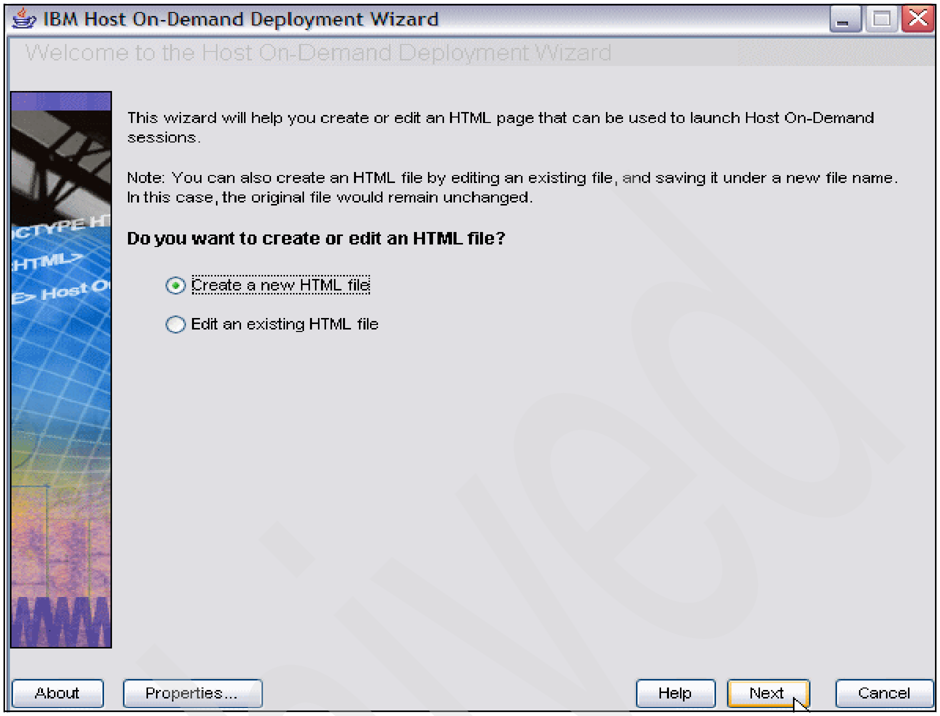
```
http://publib.boulder.ibm.com/infocenter/hodhelp/v10r0/topic/com.ibm.hod.
doc/doc/install/install12.html#install_deploywiz
```

We have added three pages under Home for each portlet, explained here:

- The DB2 on z/OS apps page will display the My Query Reports portlet.
- The ITSO HOD page will display the Host On-Demand portlet.
- The CICS Web Service page will display the CICS faces portlet.

Proceed as follows:

1. After the HOD Deployment Wizard is installed, you access it by clicking **Start** -> **All programs** -> **IBM WebSphere Host On-Demand Deployment Wizard** -> **Deployment Wizard**.
2. On the Welcome screen, select **Create a new HTML file** and click **Next**, as shown in Figure 10-2 on page 422.


IBM Host On-Demand Deployment Wizard
Welcome to the Host On-Demand Deployment Wizard

This wizard will help you create or edit an HTML page that can be used to launch Host On-Demand sessions.

Note: You can also create an HTML file by editing an existing file, and saving it under a new file name. In this case, the original file would remain unchanged.

Do you want to create or edit an HTML file?

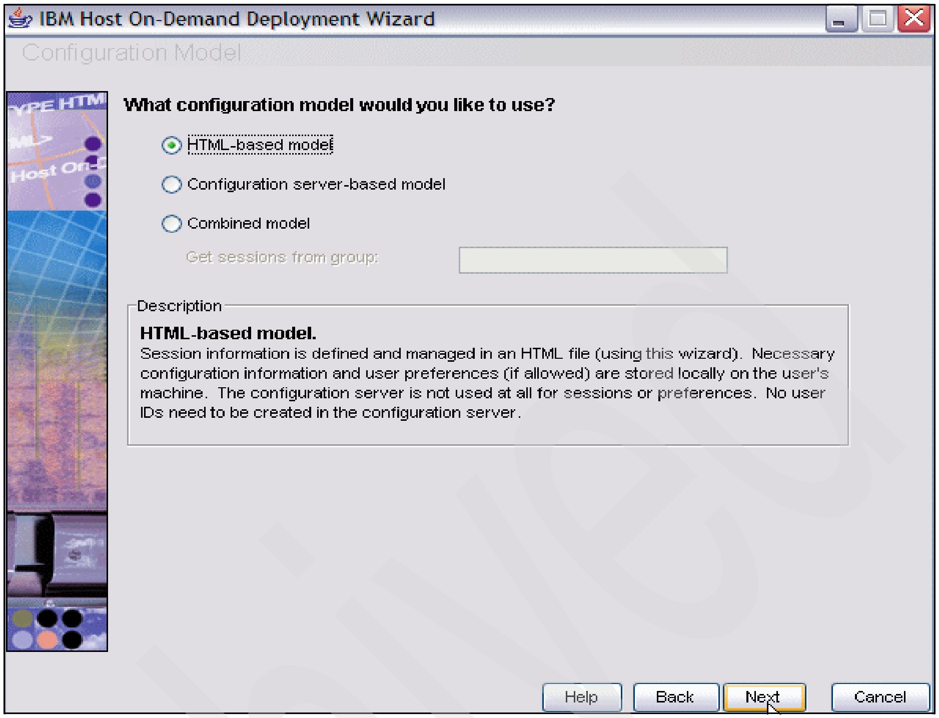
● Create a new HTML file

○ Edit an existing HTML file

About
Properties...
Help
Next
Cancel

Figure 10-2 Selecting create new HTML files

3. Select **HTML-based model** as the configuration model and click **Next**, as shown in Figure 10-3 on page 423.



IBM Host On-Demand Deployment Wizard

Configuration Model

What configuration model would you like to use?

● HTML-based model

○ Configuration server-based model

○ Combined model

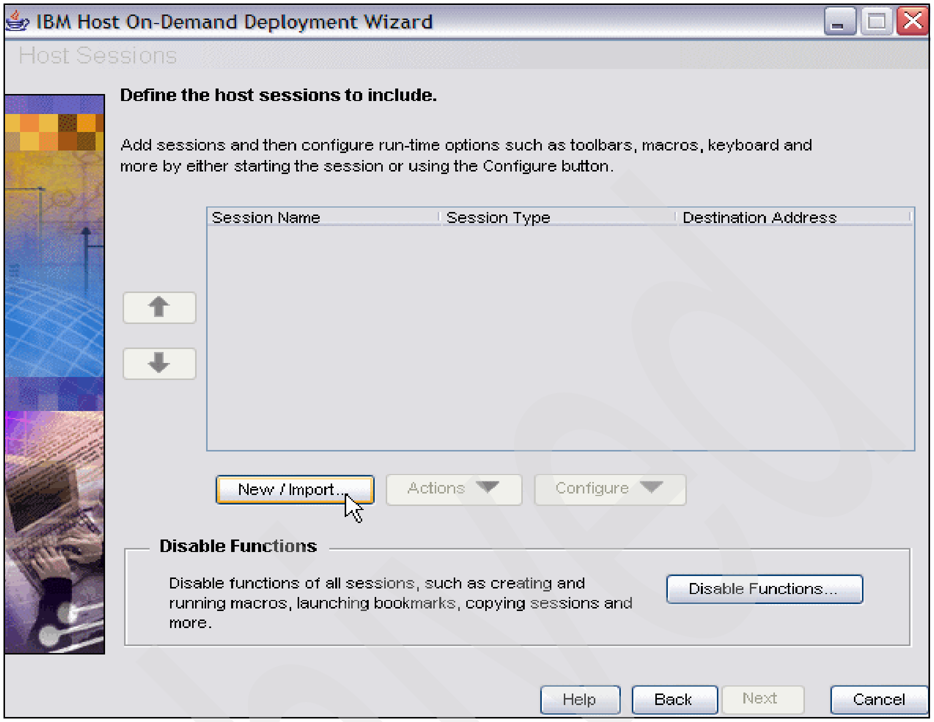
Get sessions from group:

Description

HTML-based model.
Session information is defined and managed in an HTML file (using this wizard). Necessary configuration information and user preferences (if allowed) are stored locally on the user's machine. The configuration server is not used at all for sessions or preferences. No user IDs need to be created in the configuration server.

Figure 10-3 Selecting the HTML-based configuration model

4. Click **New/Import** to create a TSO session as shown in Figure 10-4 on page 424.



Define the host sessions to include.

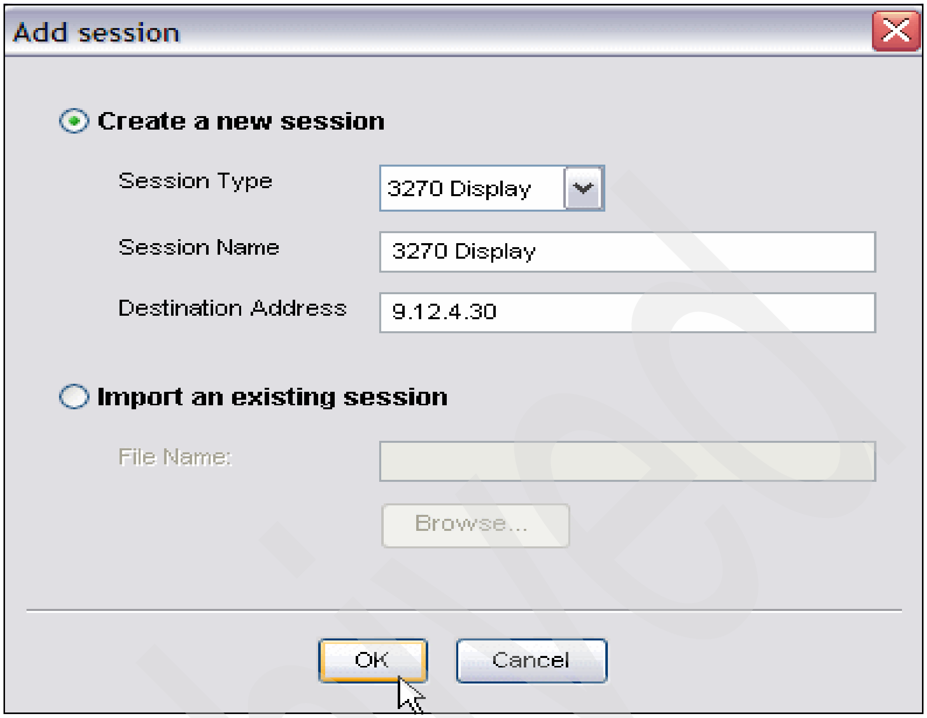
Add sessions and then configure run-time options such as toolbars, macros, keyboard and more by either starting the session or using the Configure button.

Disable Functions

Disable functions of all sessions, such as creating and running macros, launching bookmarks, copying sessions and more.

Figure 10-4 Selecting New/Import to create a TSO session

5. Select **Create a new session** and fill in the screen as shown in Figure 10-5 on page 425. Be sure to add the Destination Address of your TSO session. Click **OK** to return to the previous screen. Your TSO session should be listed.


Add session

● Create a new session

Session Type: 3270 Display

Session Name: 3270 Display

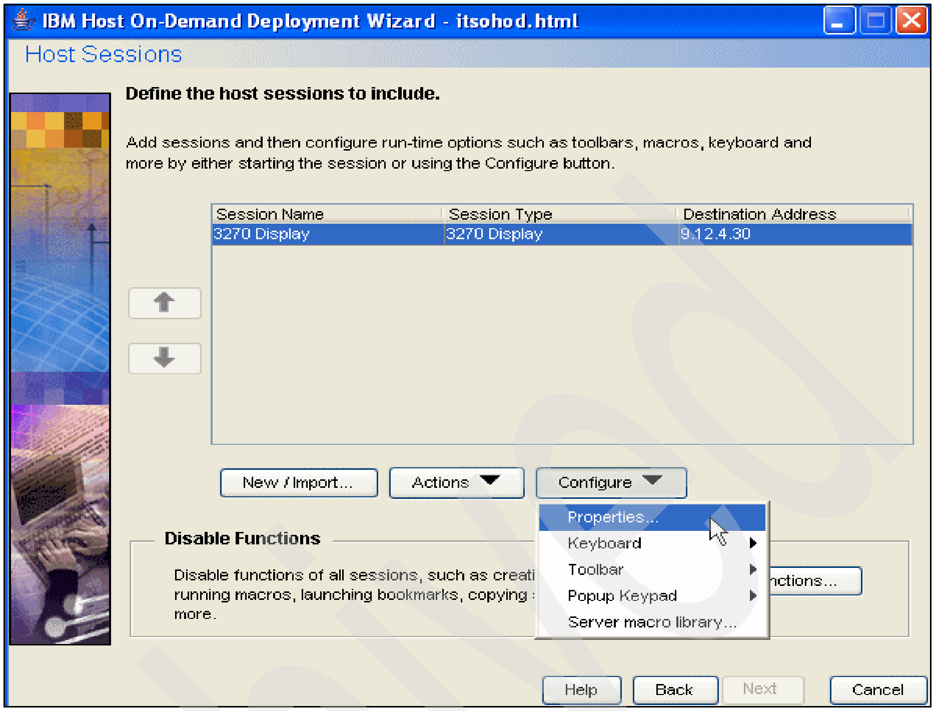
Destination Address: 9.12.4.30

○ Import an existing session

File Name:

Figure 10-5 Creating a new TSO session

6. To embed the TSO session into the portlet, you need to change the properties of this session. Click **Configure** -> **Properties** as shown in Figure 10-6 on page 426.


IBM Host On-Demand Deployment Wizard - itsohod.html

Host Sessions

Define the host sessions to include.

Add sessions and then configure run-time options such as toolbars, macros, keyboard and more by either starting the session or using the Configure button.

Session Name	Session Type	Destination Address
3270 Display	3270 Display	9.12.4.30

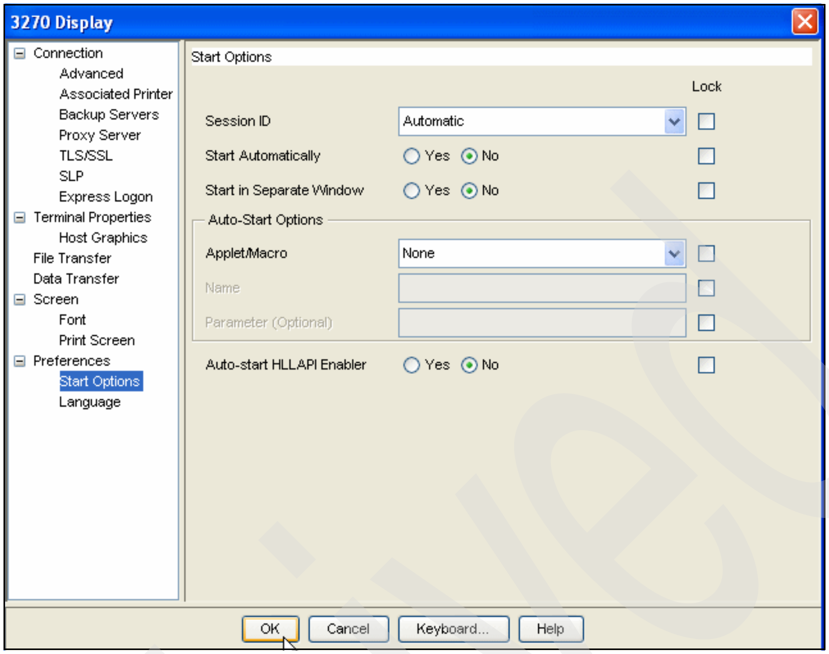
Disable Functions

Disable functions of all sessions, such as creati running macros, launching bookmarks, copying : more.

Figure 10-6   Selecting the TSO session configuration properties

7. This is the screen where you can modify the run-time options for the TSO session. Select **Preferences** -> **Start Options**. In the Start Options, change Start in Separate Window from Yes to `No` as shown in Figure 8.

8. Click **OK** to return to the previous screen and then click **Next**.



Start Options

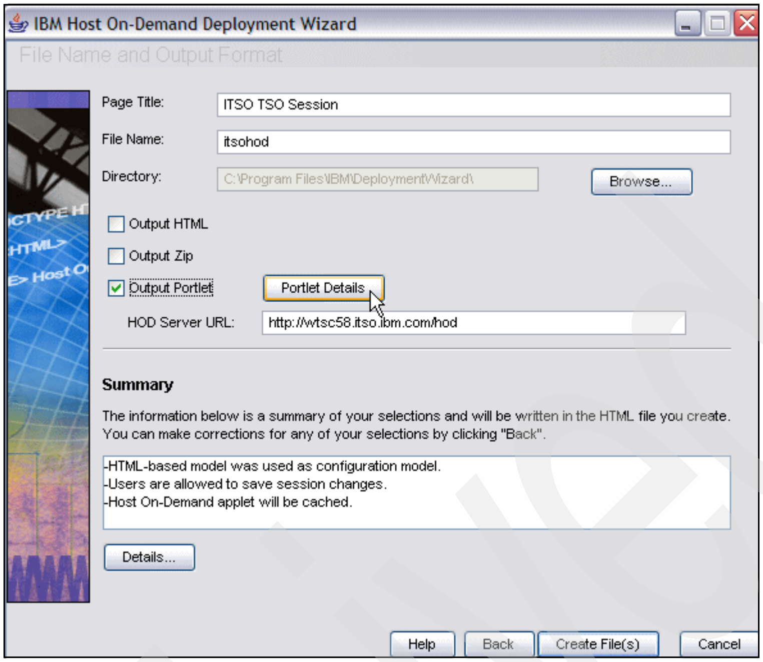
		Lock
Session ID	Automatic	□
Start Automatically	○ Yes ● No	□
Start in Separate Window	○ Yes ● No	□

Auto-Start Options

Applet/Macro	None	□
Name		□
Parameter (Optional)		□
Auto-start HLLAPI Enabler	○ Yes ● No	□

Figure 10-7 Setting the Separate Window property

9. On the Additional Options screen, accept the defaults and click **Next**.
10. On the File Name and Output Format screen as shown in Figure 10-8 on page 428, provide a Page Title, File Name, and Directory where you will find the output war file. Then uncheck Output HTML and select **Output Portlet**.
11. Click **Portlet Details** as shown in Figure 10-8 on page 428 to explore customization options. Do not make any changes. Click **OK** to return to previous screen.


IBM Host On-Demand Deployment Wizard
File Name and Output Format
Browse...
Portlet Details
Details...
Help
Back
Create File(s)
Cancel

Page Title: ITSO TSO Session

File Name: itsohod

Directory: C:\Program Files\IBM\DeploymentWizard\

□ Output HTML

□ Output Zip

☑ Output Portlet

HOD Server URL: http://wtsc58.itso.ibm.com/hod

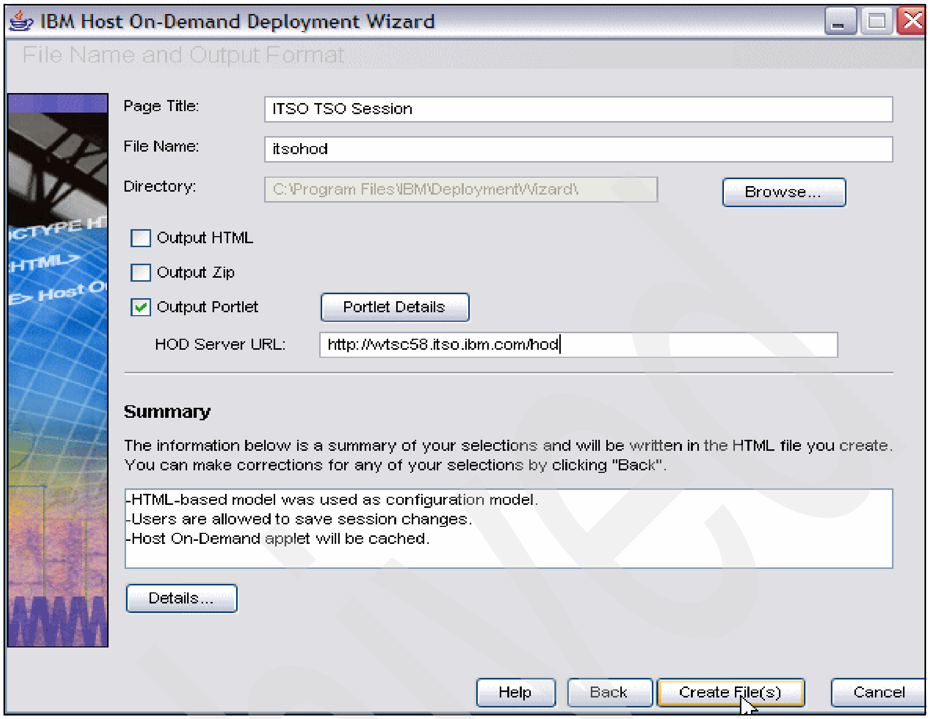
Summary

The information below is a summary of your selections and will be written in the HTML file you create. You can make corrections for any of your selections by clicking "Back".

-HTML-based model was used as configuration model.
-Users are allowed to save session changes.
-Host On-Demand applet will be cached.

Figure 10-8 Selecting the Portlet Details

12. Review the Summary section on this screen and click **Create File(s)** to create the TSO session portlet as shown in Figure 10-9 on page 429.


IBM Host On-Demand Deployment Wizard
File Name and Output Format

Page Title: ITSO TSO Session

File Name: itsohod

Directory: C:\Program Files\IBM\DeploymentWizard\

□ Output HTML

□ Output Zip

☑ Output Portlet

HOD Server URL: http://wtsc58.itso.ibm.com/hod

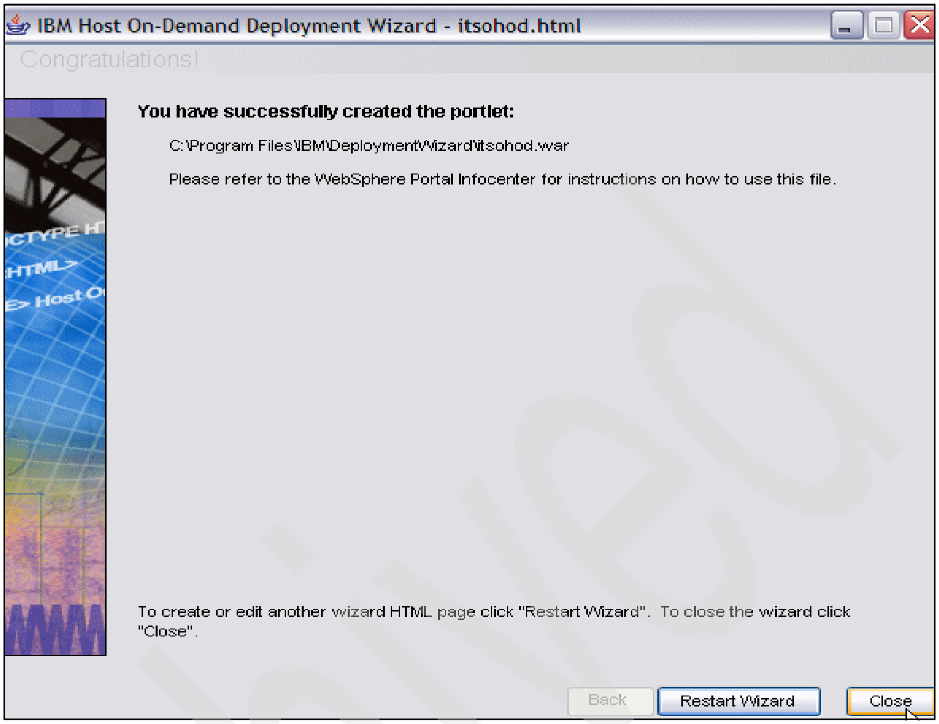
Summary

The information below is a summary of your selections and will be written in the HTML file you create. You can make corrections for any of your selections by clicking "Back".

-HTML-based model was used as configuration model.
-Users are allowed to save session changes.
-Host On-Demand applet will be cached.

Figure 10-9 Generating output files

13. On the Congratulations screen, as shown in Figure 10-10 on page 430, click **Close** to close the HOD Deployment Wizard.



You have successfully created the portlet:

C:\Program Files\IBM\DeploymentWizard\itsohod.war

Please refer to the WebSphere Portal Infocenter for instructions on how to use this file.

To create or edit another wizard HTML page click "Restart Wizard". To close the wizard click "Close".

Figure 10-10 Closing the HOD Deployment Wizard

14. Now install the HOD portlet to WebSphere Portal on z/OS. Installing a portlet is a two-step process. First select the .war file to be installed, and then view the .war file contents.

15. Open a Web browser and log in to Portal as portal administrator.

16. Click **Launch** -> **Administration** as shown in Figure 10-11 on page 431 to access the portal administration portlets.

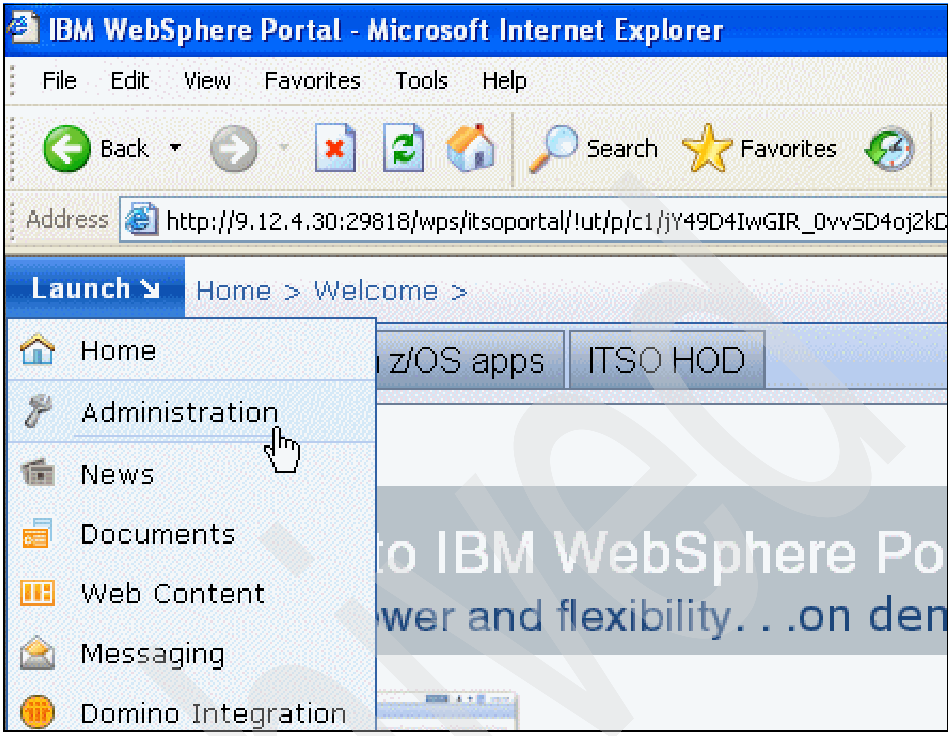

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/jY49D4IwGIR_0vvSD4oj2kD
Launch
Home > Welcome >
Home
Administration
News
Documents
Web Content
Messaging
Domino Integration
z/OS apps
ITSO HOD
to IBM WebSphere Po
wer and flexibility. . .on den

Figure 10-11 Selecting Administration

17.Click **Portlet Management** -> **Web Modules** as shown in Figure 10-12.

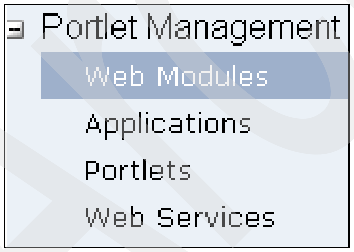
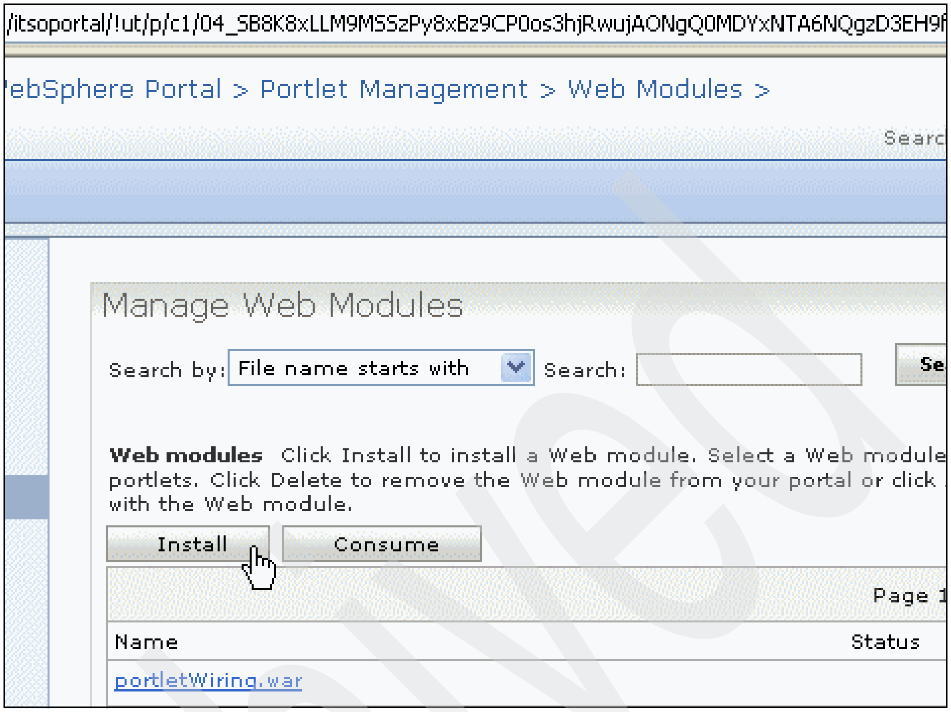


Figure 10-12 Selecting Web Modules

18.Click **Install** in Manage Web Modules as shown in Figure 10-13 on page 432.



/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjRwujAONgQ0MDYxNTA6NQgzD3EH9f

ebSphere Portal > Portlet Management > Web Modules >

Searc

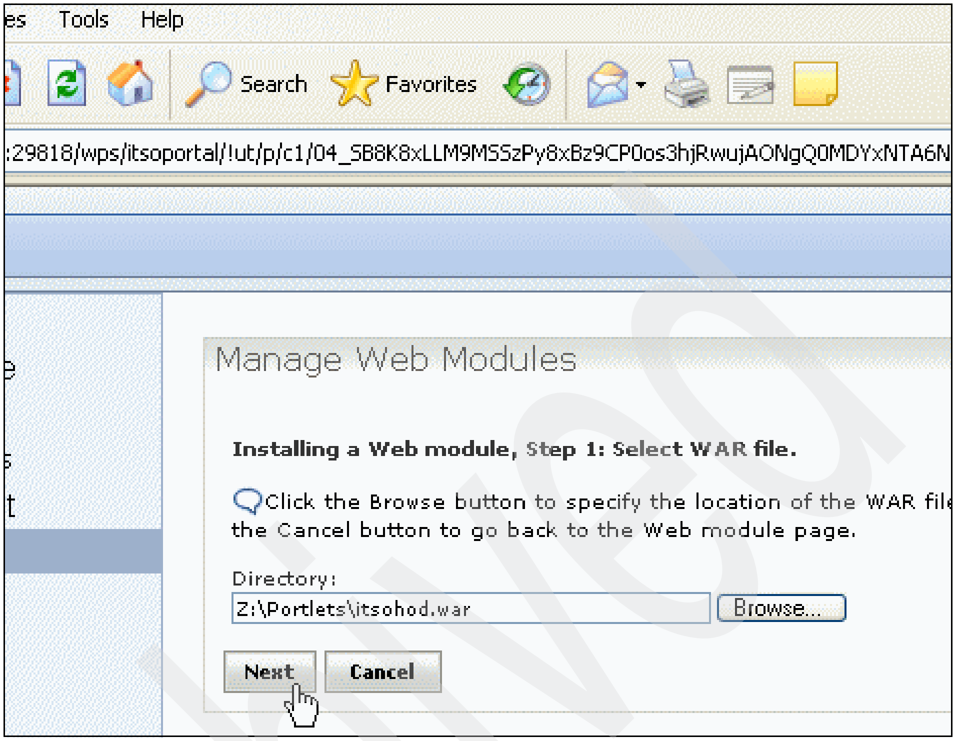
Manage Web Modules

Search by: File name starts with Search:

Web modules Click Install to install a Web module. Select a Web module portlets. Click Delete to remove the Web module from your portal or click with the Web module.

Figure 10-13 Selecting Install on Manage Web module

19. Click **Browse...** to select the itsohod.war file created by the HOD Deployment Wizard and click **Next** as shown in Figure 10-14 on page 433.


es Tools Help
Search
Favorites
:29818/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjRwujAONgQ0MDYxNTA6N
Browse...
Next
Cancel

Manage Web Modules

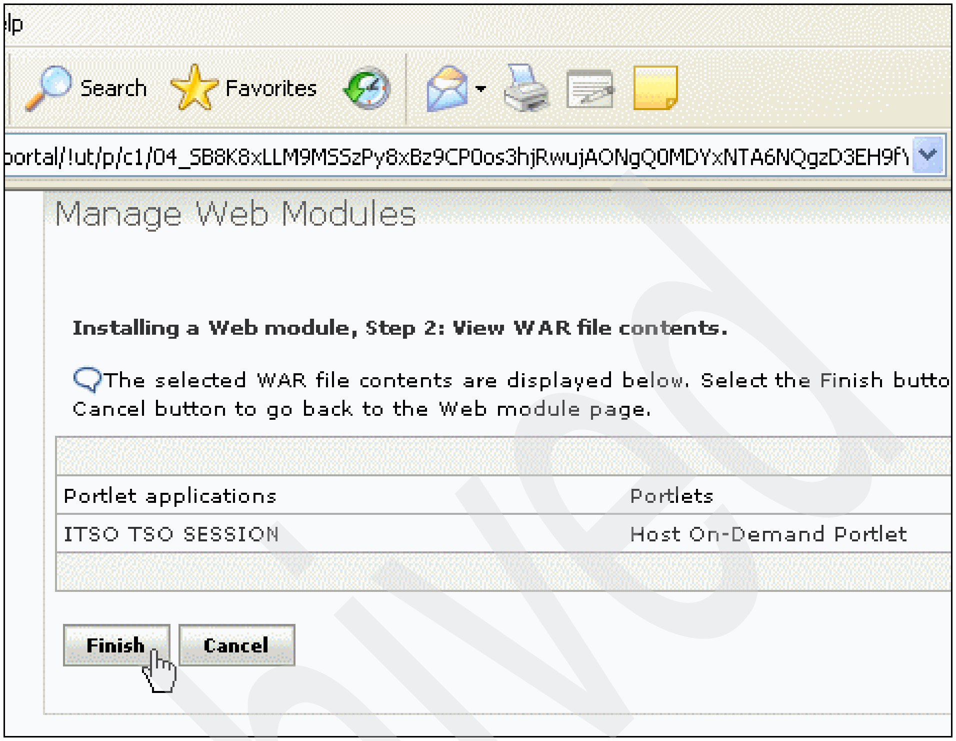
Installing a Web module, Step 1: Select WAR file.

Click the Browse button to specify the location of the WAR fil
the Cancel button to go back to the Web module page.

Directory: Z:\Portlets\itsohod.war

Figure 10-14 Manage Web Modules

20.The contents of the itsohod.war file will be displayed. Click **Finish** as shown in Figure 10-15 on page 434.


lp
Search
Favorites
portal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjRwujAONgQ0MDYxNTA6NQgzD3EH9f
Finish
Cancel

Manage Web Modules

Installing a Web module, Step 2: View WAR file contents.

The selected WAR file contents are displayed below. Select the Finish butto
Cancel button to go back to the Web module page.

Portlet applications	Portlets
ITSO TSO SESSION	Host On-Demand Portlet

Figure 10-15 Viewing the itsohod.war file contents

Be aware that the HOD portlet's war file is much bigger than 10 MB, so you will definitely have to increase the server's inbound buffer size before you can deploy the war file. Refer to “Increasing the inbound buffer size” at the following site for more information about this topic:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wcm/wcm_config_importbuffer.html`

> **Note:** If you receive message EJPAQ1330W: `Web module was successfully installed, but could not be started at this time`, then manually start the portlet application in the Manage Web Modules portlet.

21. You will now be back at the initial Manage Web Modules page. The itsohod.war should be in the list of installed Web modules.

22. To manually start (activate) the itsohod.war Web module, click ⚡.

23. After the portlet application starts successfully, add the portlet to a page. Search by Title contains and in the Search field type.

24. You should see the Java applet screen downloading the HOD Applet. After the HOD Applet is initialized, your screen should be similar to Figure 10-16.

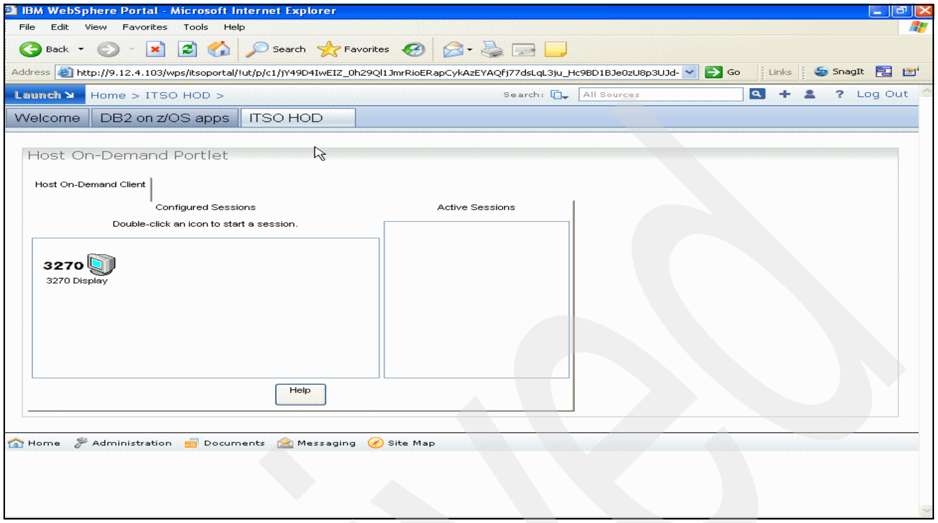
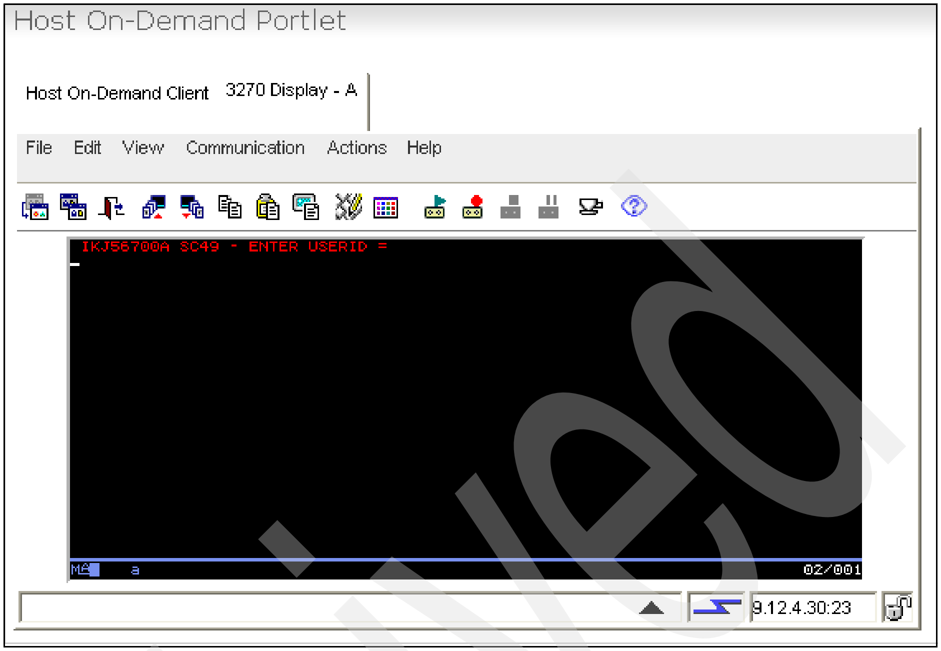


Figure 10-16 The ITSO HOD portlet

25. Double-click the 3270 icon to start a TSO session.

26. You have the same TSO session capabilities with the HOD portlet as with using a 3270 emulator.

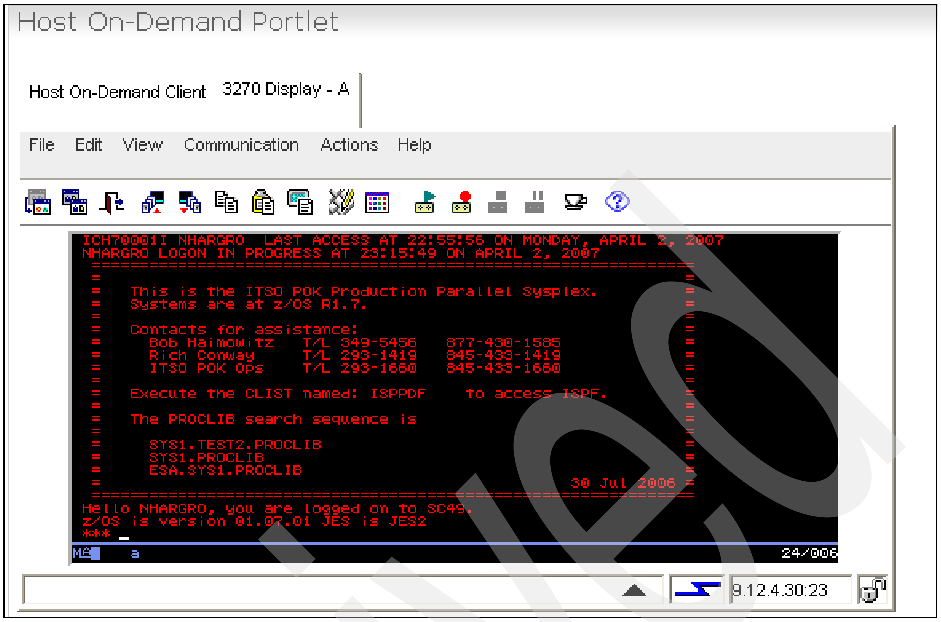
27. Figure 10-17 on page 436 through Figure 10-19 on page 437 is a series of screen shots of the HOD portlet.



```
IKJ56700A SC49 - ENTER USERID =
```

Figure 10-17 Entering user ID in TSO HOD portlet

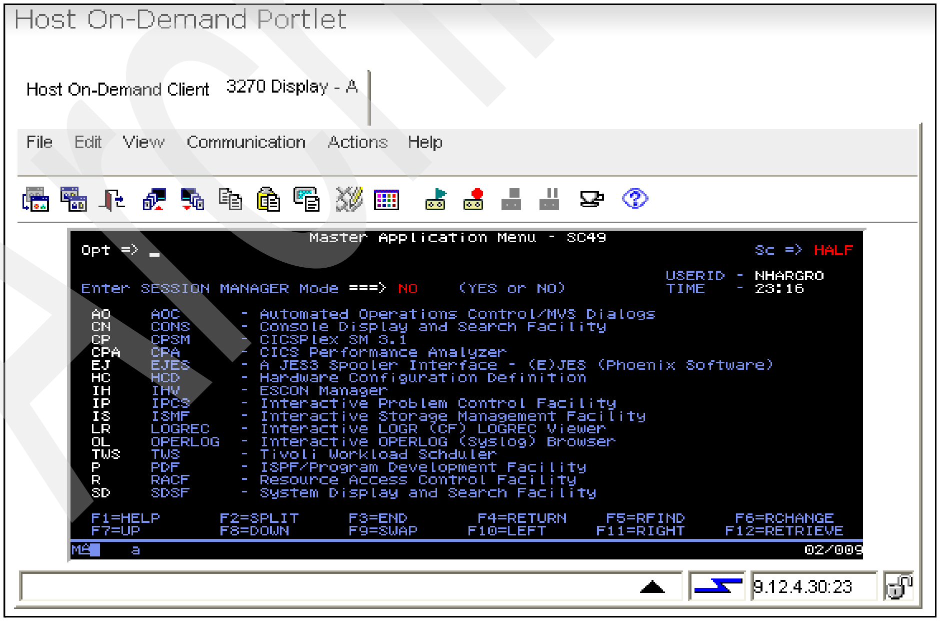
Figure 10-18 on page 437 shows the TSO logon.



```
ICH70001I NHARGRO  LAST ACCESS AT 22:55:56 ON MONDAY, APRIL 2, 2007
NHARGRO LOGON IN PROGRESS AT 23:15:49 ON APRIL 2, 2007
 ===============================================================
 =                                                             =
 =    This is the ITSO POK Production Parallel Sysplex.        =
 =    Systems are at z/OS R1.7.                                =
 =                                                             =
 =    Contacts for assistance:                                 =
 =      Bob Haimowitz    T/L 349-5456    877-430-1585          =
 =      Rich Conway      T/L 293-1419    845-433-1419          =
 =      ITSO POK Ops     T/L 293-1660    845-433-1660          =
 =                                                             =
 =    Execute the CLIST named: ISPPDF     to access ISPF.      =
 =                                                             =
 =    The PROCLIB search sequence is                           =
 =                                                             =
 =      SYS1.TEST2.PROCLIB                                     =
 =      SYS1.PROCLIB                                           =
 =      ESA.SYS1.PROCLIB                                       =
 =                                                 30 Jul 2006 =
 ===============================================================
Hello NHARGRO, you are logged on to SC49.
z/OS is version 01.07.01 JES is JES2
***
```

Figure 10-18 TSO logon in progress

Figure 10-19 shows the TSO Master Application Menu.



```
                        Master Application Menu - SC49
Opt => _                                                         Sc => HALF
                                                        USERID - NHARGRO
Enter SESSION MANAGER Mode ===> NO    (YES or NO)       TIME   - 23:16

 AO     AOC       - Automated Operations Control/MVS Dialogs
 CN     CONS      - Console Display and Search Facility
 CP     CPSM      - CICSPlex SM 3.1
 CPA    CPA       - CICS Performance Analyzer
 EJ     EJES      - A JES3 Spooler Interface - (E)JES (Phoenix Software)
 HC     HCD       - Hardware Configuration Definition
 IH     IHV       - ESCON Manager
 IP     IPCS      - Interactive Problem Control Facility
 IS     ISMF      - Interactive Storage Management Facility
 LR     LOGREC    - Interactive LOGR (CF) LOGREC Viewer
 OL     OPERLOG   - Interactive OPERLOG (Syslog) Browser
 TWS    TWS       - Tivoli Workload Schduler
 P      PDF       - ISPF/Program Development Facility
 R      RACF      - Resource Access Control Facility
 SD     SDSF      - System Display and Search Facility

 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 10-19 TSO Master Application Menu

10.2 Creating and deploying the CICS Web service client portlet application

In this section we illustrate the steps for creating a Web Service portlet application and installing the portlet application. We also demonstrate how to interact with the portlet.

Install the CICS Web service portlet to Portal on z/OS. Installing a portlet is a two-step process. First select the .war file to be installed, and then view the .war file contents, as follows:

1. Open a Web browser and log in to the Portal as Portal administrator.
2. Click **Launch -> Administration** as shown in Figure 10-20 to access the Portal Administration portlets.

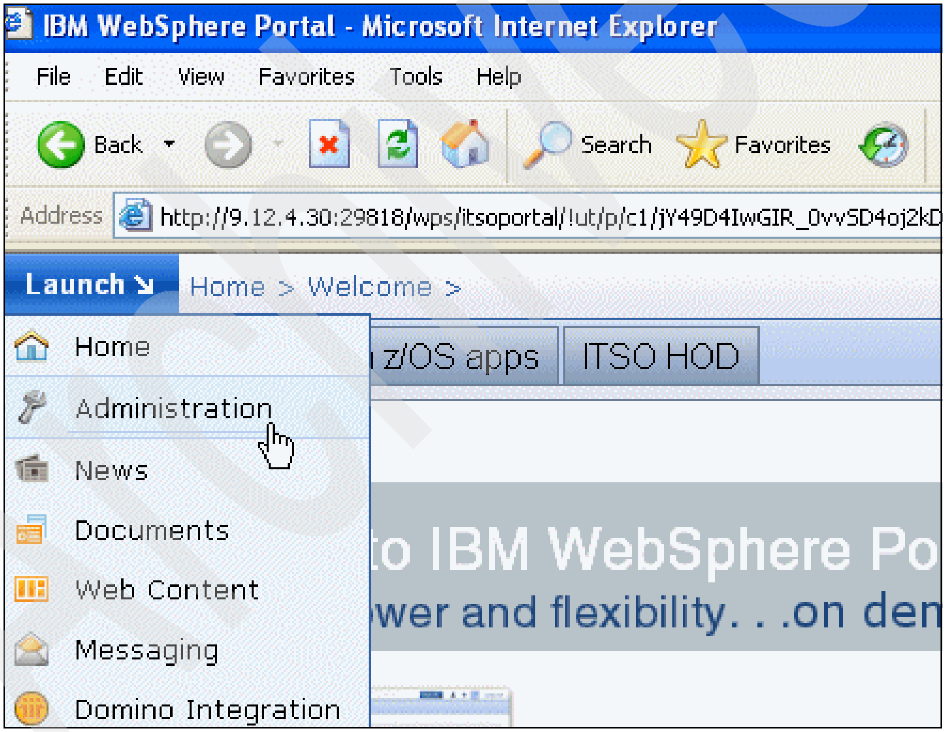


Figure 10-20 Selecting Administration

3. Click **Portlet Management -> Web Modules** as shown in Figure 10-21 on page 439.

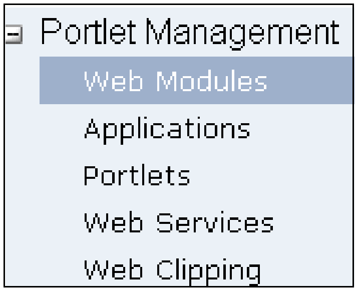
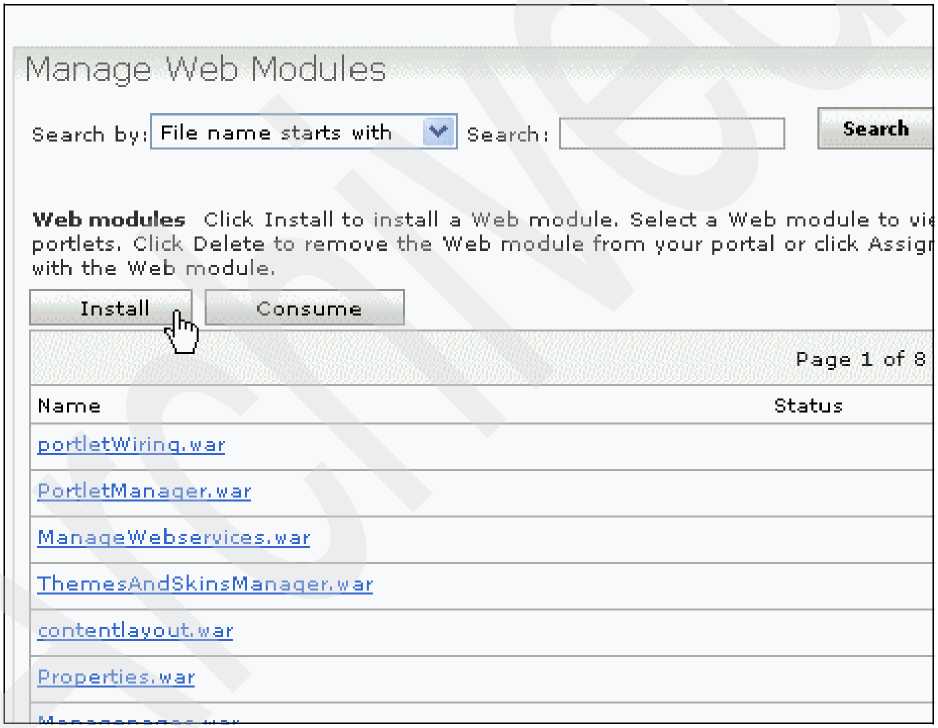


Figure 10-21 Portal Management - selecting Web Modules

4. Click **Install** in Manage Web Modules as shown in Figure 10-22.

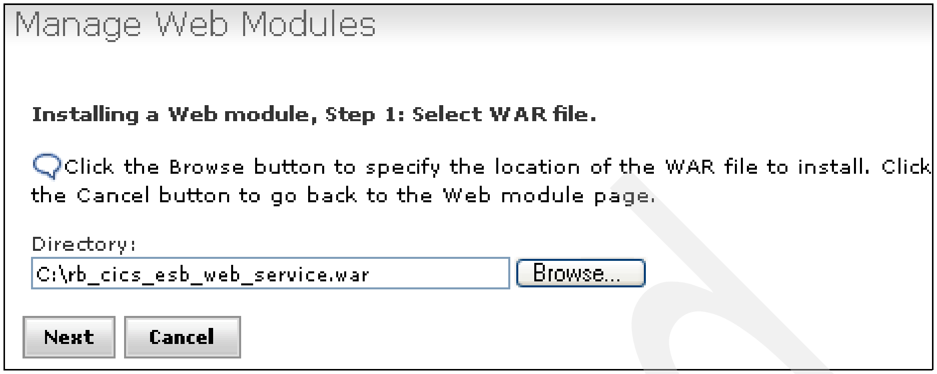


Web modules Click Install to install a Web module. Select a Web module to vie
portlets. Click Delete to remove the Web module from your portal or click Assig
with the Web module.

Name	Status
portletWiring.war	
PortletManager.war	
ManageWebservices.war	
ThemesAndSkinsManager.war	
contentlayout.war	
Properties.war	

Figure 10-22 Selecting Install on Manage Web Modules

5. Click **Browse** to select the rb_cics_esb_web_service.war exported from RAD. Then click **Next** as shown in Figure 10-23 on page 440.



Manage Web Modules

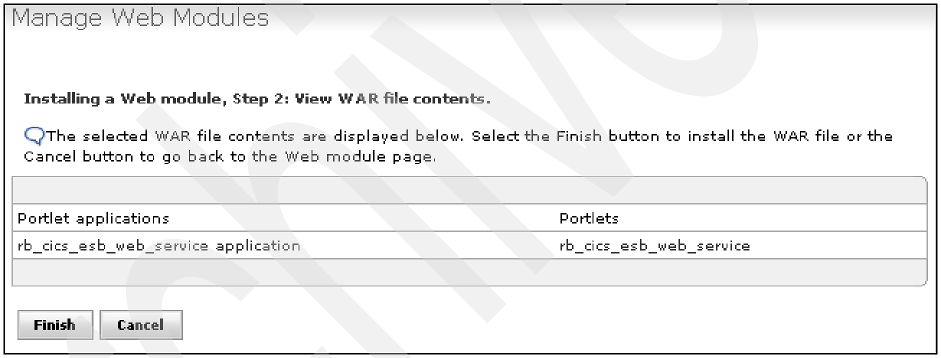
Installing a Web module, Step 1: Select WAR file.

Click the Browse button to specify the location of the WAR file to install. Click the Cancel button to go back to the Web module page.

Directory: C:\rb_cics_esb_web_service.war

Figure 10-23 Selecting CICS Web Service Web module

6. The contents of the rb_cics_esb_web_service.war file are displayed; click **Finish** as shown in Figure 10-24.



Manage Web Modules

Installing a Web module, Step 2: View WAR file contents.

The selected WAR file contents are displayed below. Select the Finish button to install the WAR file or the Cancel button to go back to the Web module page.

Portlet applications	Portlets
rb_cics_esb_web_service application	rb_cics_esb_web_service

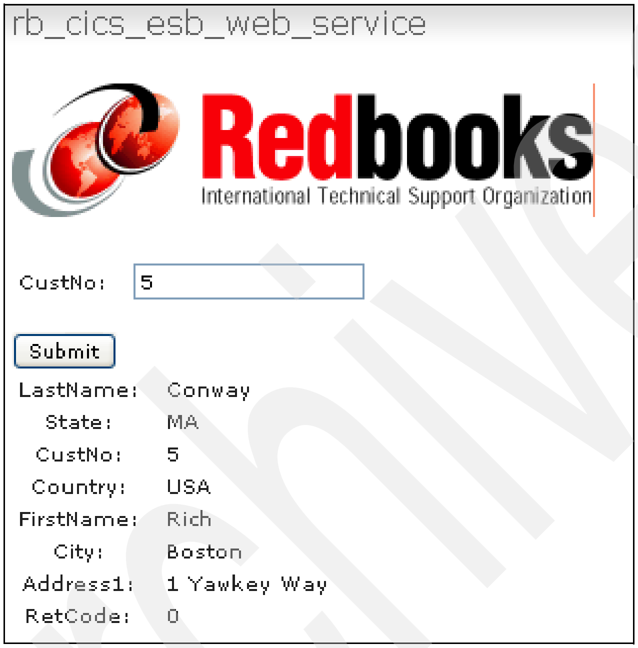
Figure 10-24 Viewing contents of CICS Web Service Web module

7. You will now be back at the initial Manage Web Modules page. The rb_cics_esb_web_serivce.war should be in the list of installed Web modules. You may have to scroll through the pages or search by File name contains to find it.

> **Note:** If you receive message EJPAQ1330W: `Web module was successfully installed, but could not be started at this time`, then manually start the portlet application in the Manage Web Modules portlet.

8. To manually start (activate) the rb_cics_esb_web_serivce.war Web module, click ⚡ .

9. After the portlet application starts successfully, add the portlet to a page. Search by Title contains and in the Search field, type `CICS`.
10. Click the page tab you created to display the CICS Web Service. In our case, it is CICS Web Service.
11. Enter any value between 1 and 10 in the CustNo. field and click **Submit**.
12. Now you should see the user information populated in the form as shown in Figure 10-25.



rb_cics_esb_web_service

CustNo:	5

LastName:	Conway
State:	MA
CustNo:	5
Country:	USA
FirstName:	Rich
City:	Boston
Address1:	1 Yawkey Way
RetCode:	0

Figure 10-25 CICS Web Service portlet

10.3 Configuring a MY Query Reports sample portlet

When Portal is installed, there are several portlets that are installed and deployed. These portlets can be found in the portal_server_root/installedApps directory.

The My Query Reports is one of those portlets that are installed and deployed with the install of Portal. This portlet displays a list of SQL queries which the user previously defined using the portlet Edit mode. When the user clicks a query, the portlet runs the query. The portlet then displays the results of the query in a table.

The user can sort the result columns by clicking the links in the table headers. The user can update, create, and test queries in the portlet Edit mode.

For more details about the My Query Reports sample portlets, refer to the following site:

`http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/portlets/myqueryreports.html`

This section explains the steps for configuring and using this portlet.

1. Create a page to display the MY Query Reports portlet. In our case, we created a page called DB2 on z/OS apps.
2. Add the My Query Reports portlet to the newly created page. Search by Title contains and in the Search field, type `Query`.
3. To configure the portlet, you must edit the shared settings. In the upper right corner of the portlet click **Portlet Menu -> Edit Share Settings** as shown in Figure 10-26.

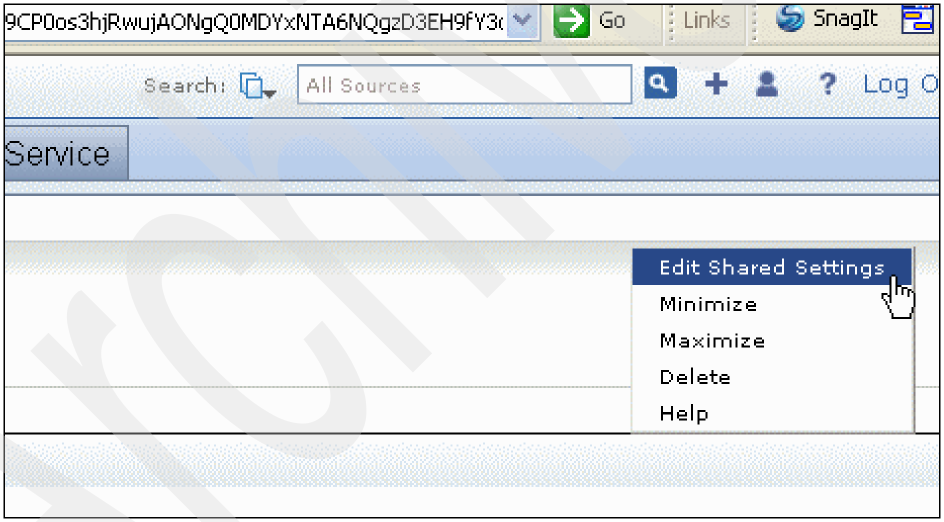


Figure 10-26 Select Edit Shared Settings

4. Figure 6 on page 444 shows the configuration view of the portlet. Click **Create** to create a query that will be saved in the list of Saved Queries.

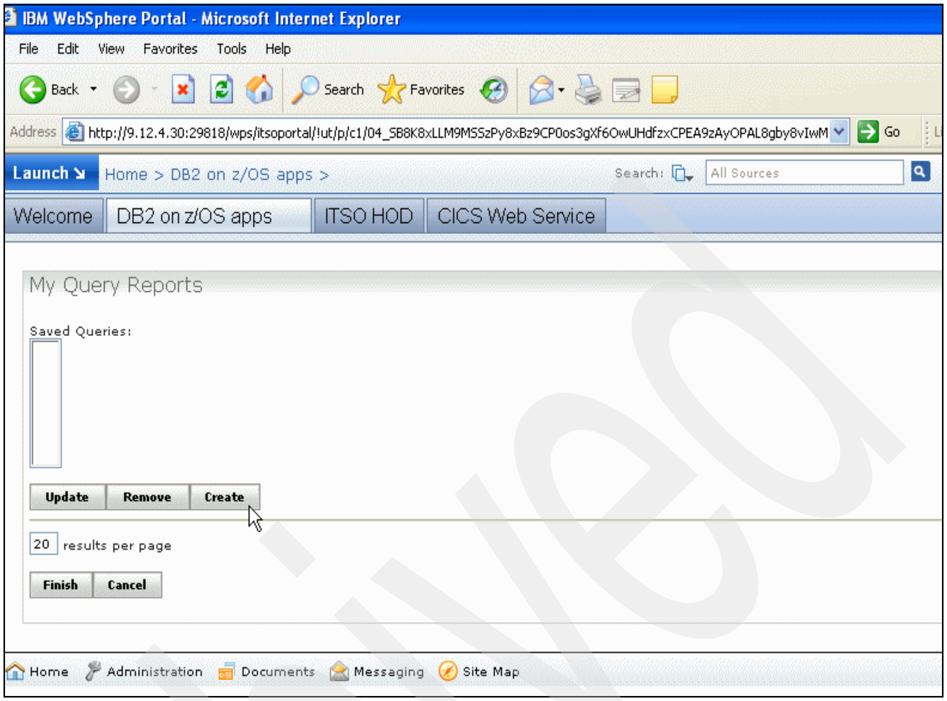

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/04_SB8K8xLLM9MSSzPy8xBz9CP0os3gXf6OwUHdfzxCPEA9zAyOPAL8gby8vIwM
Go
Launch
Home > DB2 on z/OS apps >
Search:
All Sources
Welcome
DB2 on z/OS apps
ITSO HOD
CICS Web Service
My Query Reports
Saved Queries:
Update
Remove
Create
20 results per page
Finish
Cancel
Home
Administration
Documents
Messaging
Site Map

Figure 10-27 Selecting Create to create a query

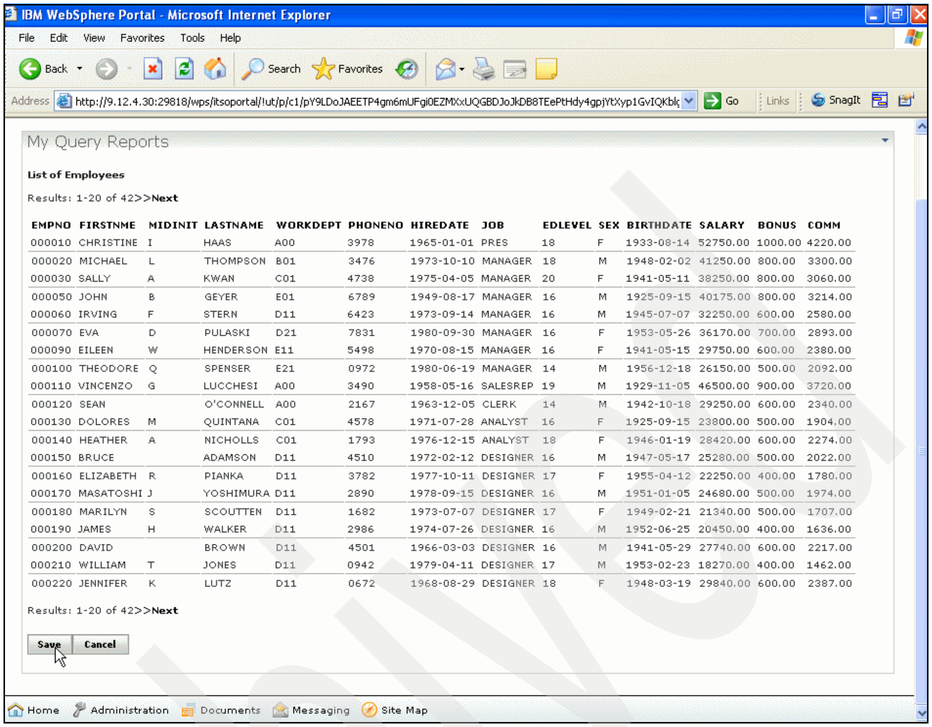
5. In the edit view of the portlet, enter the main parameters to configure a SQL query as well as the actual SQL statement. In this example, we connect to the Employee table in the DB2 Sample database that runs on z/OS using a type 4 connection, because DB2 and Portal are on the same server.

   Table 10-1 on page 444 lists SQL query settings and their description.

Table 10-1 SQL query settings

Parameter	Definition
Query Name	User provided name to call the query. This name is displayed in the bookmark link.
JDBC Driver	Name of the Java Class for the JDBC driver of your database. This driver must be in the portal's classpath.
User Name	Database user name.
Password	Database password.
Database URL	JDBC database URL i.e. jdbc:db2:sample
Column Names	Comma-separated list that will replace the original database column names when the results are displayed in a table. Skip column names by entering consecutive commas.
SQL Select Statement	The portlet only supports Select statements. You cannot update, add, or delete data in the database.

6. Fill in the SQL query settings based on your environment. Click **Test** to verify that the connection is correct and that some data is returned from the query, as shown in Figure 10-28 on page 445.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/pY9LDoJAEETP4gm6mUFgi0EZMXxUQGBDJoJkDB8TEePtHdy4gpjYtXyp1GvIQKbl
Go Links SnagIt
Home Administration Documents Messaging Site Map

My Query Reports

List of Employees

Results: 1-20 of 42>>**Next**

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES	18	F	1933-08-14	52750.00	1000.00	4220.00
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M	1948-02-02	41250.00	800.00	3300.00
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F	1941-05-11	38250.00	800.00	3060.00
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M	1925-09-15	40175.00	800.00	3214.00
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M	1945-07-07	32250.00	600.00	2580.00
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36170.00	700.00	2893.00
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29750.00	600.00	2380.00
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26150.00	500.00	2092.00
000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46500.00	900.00	3720.00
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M	1942-10-18	29250.00	600.00	2340.00
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1925-09-15	23800.00	500.00	1904.00
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28420.00	600.00	2274.00
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M	1947-05-17	25280.00	500.00	2022.00
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22250.00	400.00	1780.00
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M	1951-01-05	24680.00	500.00	1974.00
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21340.00	500.00	1707.00
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20450.00	400.00	1636.00
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M	1941-05-29	27740.00	600.00	2217.00
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18270.00	400.00	1462.00
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29840.00	600.00	2387.00

Results: 1-20 of 42>>**Next**

Figure 10-28 Results from select query

7. Click **Save** to return to the normal view of the portlet. The List of Employees query now shows up in the Save Queries list, as shown in Figure 10-29 on page 446.

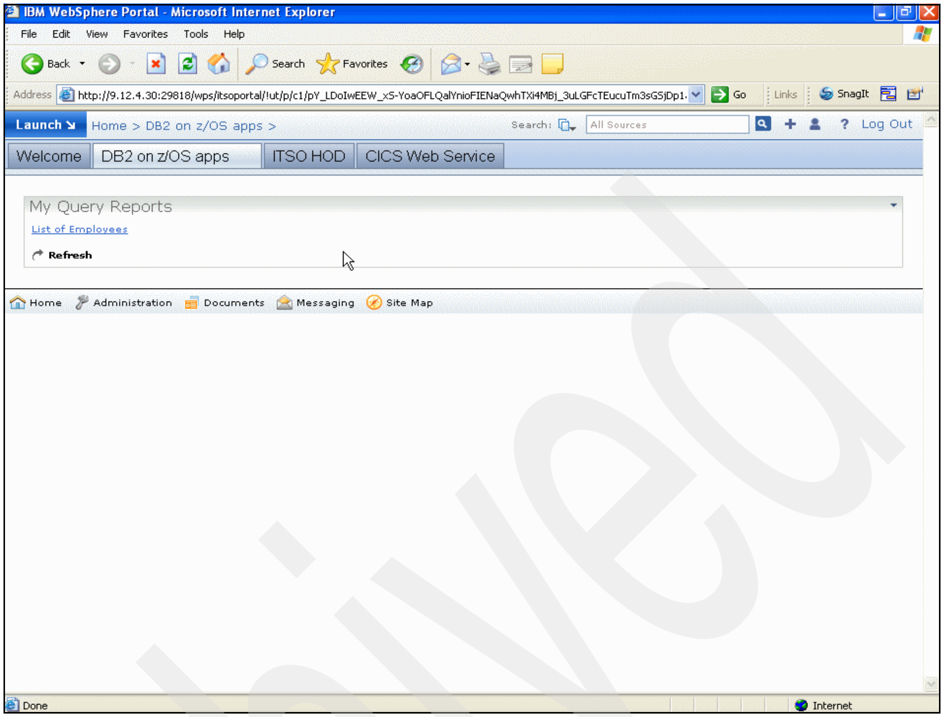

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://9.12.4.30:29818/wps/itsoportal/!ut/p/c1/pY_LDoIwEEW_x5-YoaOFLQalYnioFIENaQwhTXi4MBj_3uLGFcTEucuTm3sGSjDp1.
Go
Links
SnagIt
Launch
Home > DB2 on z/OS apps >
Search:
All Sources
Log Out
Welcome
DB2 on z/OS apps
ITSO HOD
CICS Web Service
My Query Reports
List of Employees
Refresh
Home
Administration
Documents
Messaging
Site Map
Done
Internet

Figure 10-29 Query saved in list

8. Clicking **Refresh** clears the Web Browser cache and reloads the list of queries.

Changing WebSphere Portal server ports with a jython script

In this appendix we show an example of a jython script that we used to change the WebSphere Portal server ports (Example A-1) and a JCL to run this script (Example A-2 on page 456).

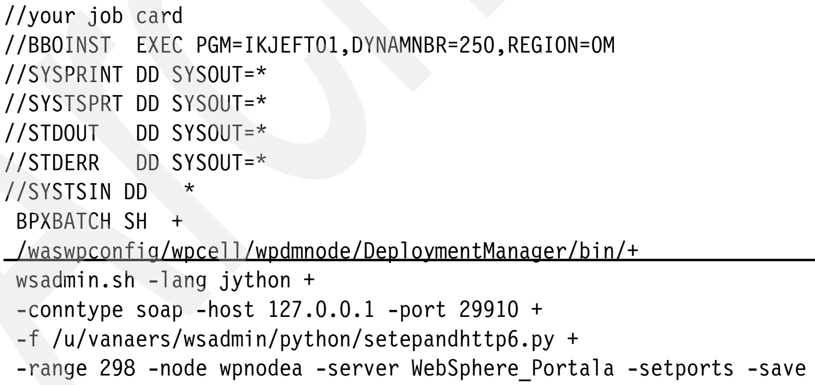
Example A-1 Jython script for changing the WebSphere Portal server ports

```
#----------------------------------------------------------------------------------------
#                 setepandhttp6.py
#
#   this script can display and set the endpoints of an application server  ------------
#   set depends on parameter  -setports (display is default)
#   dry run (no final save) if parameter -save NOT specified
#   parameters
#     -setports        set endpoints  (default display)
#     -save            save configuration (default nosave)
#    ---> for PSSC Wildfire
#     - team(nr)       11, 12, 21, 22, 31, 32, 41, 42 -
#                      (portnr allways starts with teamnr + (see logic))
#    ---> for ITSOPOK
#     -range           (3 digits) start of portnr - 2 digits are added see logic
#     -node            nodename of the node where the server resides
#     -server          servername
#----------------------------------------------------------------------------------------
import sys
def wsadminToList(inStr):
        outList=[]
        if (len(inStr)>0 and inStr[0]=='[' and inStr[-1]==']'):
                tmpList = inStr[1:-1].split(" ")
        else:
                tmpList = inStr.split("\n")  #splits for Windows or Linux
        for item in tmpList:
                item = item.rstrip();        #removes any Windows "\r"
                if (len(item)>0):
                        outList.append(item)
        return outList
#endDef
#---------------------------------------------------------------------------------------
#   this file allows the change of the Endpoints of an Application Server
#   the jacl file can be executed with connection SOAP
#                                                 NONE
#---------------------------------------------------------------------------------------
print "->setepandhttp6 PSSC JYTHON version 6.1"
argerr = 0
errstr = ""
newPort = ""
i = 0  #forStart
argc = len(sys.argv)
while ( i < argc ):  #forTest
        arg = sys.argv[i]
        #set j [expr {${i} + 1}]
        #puts "3"
        #set arg2 [lindex $argv $j]
        #puts "-----argument $i  $arg   $j  $arg2 "
        if (arg == "-server"):
                i = i+1;
                if (i < argc):
                        serverName = sys.argv[i]
                else:
                        argerr = 2
                #endElse
        elif (arg == "-node"):
                i = i+1;
                if (i < argc):
                        nodeName = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-teamnr"):
                i = i+1;
                if (i < argc):
                        teamnr = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-range"):
                i = i+1;
                if (i < argc):
                        range = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-team"):
                i = i+1;
                if (i < argc):
                        teamnr = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-servnr"):
                i = i+1;
                if (i < argc):
                        servnr = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-boot"):
                i = i+1;
                if (i < argc):
                        bootPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-http"):
                i = i+1;
                if (i < argc):
                        httpPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-https"):
                i = i+1;
                if (i < argc):
                        httpsport = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-iiop"):
                i = i+1;
                if (i < argc):
                        iiopPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-iiops"):
                i = i+1;
                if (i < argc):
                        iiopsPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-drs"):
                i = i+1;
                if (i < argc):
                        drsPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-soap"):
                i = i+1;
                if (i < argc):
                        soapPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-jmsq"):
                i = i+1;
                if (i < argc):
                        jmsqPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-jmsd"):
                i = i+1;
                if (i < argc):
                        jmsdPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-sib"):
                i = i+1;
                if (i < argc):
```

```
                        sibPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-sibs"):
                i = i+1;
                if (i < argc):
                        sibsPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-sibm"):
                i = i+1;
                if (i < argc):
                        sibmPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-sibms"):
                i = i+1;
                if (i < argc):
                        sibmsPort = sys.argv[i]
                else:
                        argerr = 3
                #endElse
        elif (arg == "-setports"):
                setports = "YES"
        elif (arg == "-save"):
                savehere = "YES"
        else:
                argerr = 10
        #endElse
        i += 1  #forNext
#endWhile  (#endFor)
if (argerr > 0):
        print "->setepandhttp6 ending with error "+argerr+" for "+errstr
        sys.exit()
#endIf
cellId = AdminConfig.getid("/Cell:/")
cellName = AdminConfig.showAttribute(cellId, "name")
if ( not  ("servnr" in locals().keys() or "servnr" in globals().keys())  ):
        servnr = "1"
#endIf
print "->setepandhttp6 parameters seem to be OK"
if ( ("teamnr" in locals().keys() or "teamnr" in globals().keys())  ):
        teamstr = "Team"
        teamstr += teamnr
        if ( not  ("dmgr" in locals().keys() or "dmgr" in globals().keys())  ):
        if ( not  ("serverName" in locals().keys() or "serverName" in globals().keys())  ):
                serverName = teamstr
                serverName += "Serv0"
                serverName += servnr
        #endIf
        if ( not  ("nodeName" in locals().keys() or "nodeName" in globals().keys())  ):
                nodeName = teamstr
                nodeName += "Node01"
        #endIf
    else:
        serverName = "dmgr"
                nodeName = teamstr
                nodeName += "DmgrNode"
                setports = ""
    #endif
else:
    if ( not  ("serverName" in locals().keys() or "serverName" in globals().keys())  ):
                print "->setepandhttp6 serverName mandatory if NO teamnr"
        sys.exit()
        #endIf
        if ( not  ("nodeName" in locals().keys() or "nodeName" in globals().keys())  ):
                print "->setepandhttp6 nodeName mandatory if NO teamnr"
        sys.exit()
        #endIf
#endIf
servid = AdminConfig.getid("/Cell/"+cellName+"/Node/"+nodeName+"/Server/"+serverName )
if (servid == ""):
        print "->setepandhttp6 invalid Server "+serverName+" in node("+nodeName+")
cell("+cellName+")"
        sys.exit()
#endIf
print "->setepandhttp6 for server("+serverName+") in node("+nodeName+") cell("+cellName+")"
if ( ("teamnr" in locals().keys() or "teamnr" in globals().keys())  ):
        print "->setepandhttp6 ports will be changed for team " + teamnr
        if (teamnr == "91"):
                teamst = "51"
        elif (teamnr == "92"):
                teamst = "52"
        elif (teamnr == "93"):
                teamst = "53"
        elif (teamnr == "94"):
                teamst = "54"
        else:
                teamst = teamnr
        #endIf
        saveteam = teamst
        if ( ("setports" in locals().keys() or "setports" in globals().keys())  ):
            teamst += servnr
        httpPort = teamst
        httpPort += "0"
        httpsPort = teamst
        httpsPort += "1"
                bootPort = teamst
                bootPort += "2"
                iiopPort = teamst
                iiopPort += "2"
                iiopsPort = teamst
                iiopsPort += "3"
                soapPort = teamst
                soapPort += "4"
                drsPort = teamst
                drsPort += "5"
                jmsqPort = teamst
                jmsqPort += "62"
                jmsdPort = teamst
                jmsdPort += "61"
                dcsPort = teamst
                dcsPort += "7"
                sibPort = teamst
                sibPort += "81"
                sibsPort = teamst
                sibsPort += "82"
                sibmPort = teamst
                sibmPort += "83"
                sibmsPort = teamst
                sibmsPort += "84"
                sasslPort = teamst
                sasslPort += "85"
                WC_admPort = teamst
                WC_admPort += "01"
                WC_admsPort = teamst
                WC_admsPort += "11"
                WC_defPort = teamst
                WC_defPort += "0"
                WC_defsPort = teamst
                WC_defsPort += "1"
                CServAPort = teamst
                CServAPort += "95"
                CMutlAPort = teamst
                CMutlAPort += "96"
                SIPPort = teamst
                SIPPort += "02"
                SIPPorts = teamst
                SIPPorts += "12"
        #endIf
elif ( ("range" in locals().keys() or "range" in globals().keys())  ):
    if ( ("setports" in locals().keys() or "setports" in globals().keys())  ):
        print "->setepandhttp6 ports will be changed to range " +range
        httpPort = range
        httpPort += "18"
        httpsPort = range
        httpsPort += "19"
        bootPort = range
        bootPort += "11"
        iiopPort = range
                iiopPort += "11"
                iiopsPort = range
                iiopsPort += "12"
                soapPort = range
                soapPort += "10"
                drsPort = range
                drsPort += "13"
                jmsqPort = range
                jmsqPort += "16"
                jmsdPort = range
                jmsdPort += "17"
                dcsPort = range
                dcsPort += "15"
                sibPort = range
                sibPort += "20"
                sibsPort = range
                sibsPort += "21"
                sibmPort = range
                sibmPort += "22"
                sibmsPort = range
                sibmsPort += "23"
                sasslPort = range
                sasslPort += "24"
                WC_admPort = range
                WC_admPort += "28"
                WC_admsPort = range
                WC_admsPort += "29"
                WC_defPort = range
                WC_defPort += "18"
                WC_defsPort = range
                WC_defsPort += "19"
                CServAPort = range
                CServAPort += "24"
                CMutlAPort = range
                CMutlAPort += "25"
                SIPPort = range
                SIPPort += "26"
                SIPPorts = range
                SIPPorts += "27"
else:
    if ( ("setports" in locals().keys() or "setports" in globals().keys())  ):
        print "->setepandhttp6 individual ports will be changed"
        #endIf
#endIf

i = 0
#puts "**** [$AdminConfig showall $servid]"
#--------HTTP(S)_ADDRESSES-----------
```

```
wc = AdminConfig.list("WebContainer", servid )
transportAttr = AdminConfig.showAttribute(wc, "transports" )
transports = transportAttr[1:len(transportAttr)-1].split(" ")
for transport in transports:
    #print "->setepandhttp6 2" + transport
        address = AdminConfig.showAttribute(transport, "address" )
        #print "->setepandhttp6 3" + address
        port = AdminConfig.showAttribute(address, "port")
        host = AdminConfig.showAttribute(address, "host")
        ssl = AdminConfig.showAttribute(transport,"sslEnabled")
        print "-> *** HTTP_ADDRESS current port "+port+" on host "+host+" sslenabled "+ssl
        if (ssl == "true"):
                addresshttps = address
        else:
                addresshttp = address
        #endElse
#endFor
if (host == ""):
        host = "*"
#endIf
if ( ("setports" in locals().keys() or "setports" in globals().keys())  ):
        if ( ("httpPort" in locals().keys() or "httpPort" in globals().keys())  ):
                print "-> ***-> HTTP_ADDRESS new port "+httpPort+" on host "+host
                AdminConfig.modify(addresshttp, [["host", host], ["port", httpPort]] )
        #endIf
        if ( ("httpsPort" in locals().keys() or "httpsPort" in globals().keys())  ):
                print "-> ***-> HTTPS_ADDRESS new port "+httpsPort+" on host "+host
                AdminConfig.modify(addresshttps, [["host", host], ["port", httpsPort]] )
        #endIf
#endIf
#------------------------------------
print "-> *** EndPoint Addresses ****"
#----------------------------------------------------------------- -----------
nodeid = AdminConfig.getid("/Cell/"+cellName+"/Node/"+nodeName )
serverEntriesAttr = (AdminConfig.list("ServerEntry", nodeid))
#print "###1" + serverEntriesAttr +"/"
serverEntries = serverEntriesAttr.split(java.lang.System.getProperty("line.separator"))
for serverEntry in serverEntries:
    #print "###1A" + serverEntry +"/"
        sName = AdminConfig.showAttribute(serverEntry, "serverName" )
        #print "###1B" + sName +"/"
        if (sName == serverName):
                print " ***-> Special entries for server "+sName
                specialEndPoints = AdminConfig.showAttribute(serverEntry, "specialEndpoints" )
                #print "###1BA" + specialEndPoints +"/"
                specialEndPoints = AdminConfig.showAttribute(serverEntry,
"specialEndpoints")[1:len(specialEndPoints)-1].split(" ")
                for specialEndPoint in specialEndPoints:
                        endPointNm = AdminConfig.showAttribute(specialEndPoint, "endPointName" )
                        #print "###1BAA" + endPointNm
                        newPort = ""
                        epoint = AdminConfig.showAttribute(specialEndPoint, "endPoint" )
                        port = AdminConfig.showAttribute(epoint,"port")
                        host = AdminConfig.showAttribute(epoint,"host")
                        print "   "+endPointNm+"  current port "+port+" on host "+host+" "
                        if ( ("setports" in locals().keys() or "setports" in globals().keys())  ):
                                if (endPointNm == "JMSSERVER_DIRECT_ADDRESS"):
                                        if ( ("jmsdPort" in locals().keys() or "jmsdPort" in
globals().keys())  ):
                                                newPort = jmsdPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "JMSSERVER_QUEUED_ADDRESS"):
                                        if ( ("jmsqPort" in locals().keys() or "jmsqPort" in
globals().keys())  ):
                                                newPort = jmsqPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "ORB_LISTENER_ADDRESS"):
                                        if ( ("iiopPort" in locals().keys() or "iiopPort" in
globals().keys())  ):

                                                newPort = iiopPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "ORB_SSL_LISTENER_ADDRESS"):
                                        if ( ("iiopsPort" in locals().keys() or "iiopsPort" in
globals().keys())  ):

                                                newPort = iiopsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SOAP_CONNECTOR_ADDRESS"):
                                        if ( ("soapPort" in locals().keys() or "soapPort" in
globals().keys())  ):

                                                newPort = soapPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIB_ENDPOINT_ADDRESS"):
                                        if ( ("sibPort" in locals().keys() or "sibPort" in
globals().keys())  ):

                                                newPort = sibPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIB_ENDPOINT_SECURE_ADDRESS"):
                                        if ( ("sibsPort" in locals().keys() or "sibsPort" in
globals().keys())  ):

                                                newPort = sibsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIB_MQ_ENDPOINT_ADDRESS"):
                                        if ( ("sibmPort" in locals().keys() or "sibmPort" in
globals().keys())  ):

                                                newPort = sibmPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIB_MQ_ENDPOINT_SECURE_ADDRESS"):
                                        if ( ("sibmsPort" in locals().keys() or "sibmsPort" in
globals().keys())  ):

                                                newPort = sibmsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "BOOTSTRAP_ADDRESS"):
                                        if ( ("bootPort" in locals().keys() or "bootPort" in
globals().keys())  ):

                                                newPort = bootPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "DCS_UNICAST_ADDRESS"):
                                        if ( ("dcsPort" in locals().keys() or "dcsPort" in
globals().keys())  ):

                                                newPort = dcsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SAS_SSL_SERVERAUTH_LISTENER_ADDRESS"):
                                        if ( ("sasslPort" in locals().keys() or "sasslPort" in 
globals().keys())  ):
                                                newPort = sasslPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "WC_adminhost"):
                                        if ( ("WC_admPort" in locals().keys() or "WC_admPort" in 
globals().keys())  ):
                                                newPort = WC_admPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "WC_adminhost_secure"):
                                        if ( ("WC_admsPort" in locals().keys() or "WC_admsPort" in 
globals().keys())  ):
                                                newPort = WC_admsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "WC_defaulthost"):
                                        if ( ("WC_defPort" in locals().keys() or "WC_defPort" in 
globals().keys())  ):
                                                newPort = WC_defPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "WC_defaulthost_secure"):
                                        if ( ("WC_defsPort" in locals().keys() or "WC_defsPort" in 
globals().keys())  ):
                                                newPort = WC_defsPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS"):
                                        if ( ("CServAPort" in locals().keys() or "CServAPort" in 
globals().keys())  ):
                                                newPort = CServAPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS"):
                                        if ( ("CMutlAPort" in locals().keys() or "CMutlAPort" in 
globals().keys())  ):
                                                newPort = CMutlAPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIP_DEFAULTHOST"):
                                        if ( ("SIPPort" in locals().keys() or "SIPPort" in 
globals().keys())  ):
                                                newPort = SIPPort
                                        else:
                                                newPort = ""
                                        #endElse
                                elif (endPointNm == "SIP_DEFAULTHOST_SECURE"):
                                        if ( ("SIPPorts" in locals().keys() or "SIPPorts" in 
globals().keys())  ):
                                                newPort = SIPPorts
                                        else:
                                                newPort = ""
                                        #endElse
                                else:
                                        print "->setepandhttp6 bypass "+endPointNm+" NOT considered 
<-"
                                        newPort = ""
                                #endElse
                                if ( ("newPort" in locals().keys() or "newPort" in globals().keys())
):
                                        if (newPort == port):
                                                print "->setepandhttp6 bypass change <-"
                                        elif (newPort == ""):
                                                print "->setepandhttp6 NO new value for
"+endPointNm+" bypass"
                                        else:
                                                print "->setepandhttp6 ****** "+endPointNm+" old
port("+port+") new port("+newPort+") on host("+host+") <-"
                                                AdminConfig.modify(epoint, [["host", host], ["port",
newPort]] )
                                                print "->setepandhttp6 change OK <-"
                                        #endElse
                                #endIf
                        #endIf
                #endFor
        #endIf
#endFor
if ( ("savehere" in locals().keys() or "savehere" in globals().keys())  ):
        print "->setepandhttp6 saving"
        try:
        print "->setepandhttp6 try to save with synclevel"
        sync1 = AdminControl.completeObjectName("type=NodeSync,node="+nodeName+",*" )
        print "->psscserver6 saving with synclevel "+sync1+" "
        AdminControl.invoke(sync1, "sync" )
        print "->setepandhttp6 synclevel accepted"
        except:
        print "->setepandhttp6 Unexpected error:", sys.exc_info()[0]
        print "->setepandhttp6 synclevel denied"
        else:
            AdminConfig.save( )
        print "->setepandhttp6 after save"
#endIf
print "***----HTTP(S) and EndPoint display/settings ENDED(JYTHON)-------------------"
```

Example A-2   Sample JCL for running the port change jython script



```
//your job card
//BBOINST  EXEC PGM=IKJEFT01,DYNAMNBR=250,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSTSIN DD   *
 BPXBATCH SH  +
 /waswpconfig/wpcell/wpdmnode/DeploymentManager/bin/+
 wsadmin.sh -lang jython +
 -conntype soap -host 127.0.0.1 -port 29910 +
 -f /u/vanaers/wsadmin/python/setepandhttp6.py +
 -range 298 -node wpnodea -server WebSphere_Portala -setports -save
```

B

Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247459`

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 IBM Redbook form number, SG247459.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
cics_service.zip	Portal application calling a CICS Web Service

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks” on page 460. Note that some of the documents referenced here may be available in softcopy only.

- *Distributed Security and High Availability with Tivoli Access Manager and WebSphere Application Server for z/OS*, SG24-6760
- *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850

Other publications

These publications are also relevant as further information sources:

- *Tivoli Directory Server for z/OS V1R8.0 Administration and Use,* SC23-5191
- *z/OS LDAP Server Administration and Use*, SC24-5923

Online resources

These Web sites are also relevant as further information sources:

- Supported hardware and software for WebSphere® Portal V6.0:

  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/inst_req60.html\`

- Software corequisites can be found in the supported hardware and software section of the IBM WebSphere Portal Enable for z/OS Version 6.0.0.1 Infocenter site:

  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.zos.doc/wpf/inst_req60.html`

- For more information about the use of variables and application profiles, refer to “Tips for using the customization dialog” in the WebSphere Portal for z/OS Infocenter at the following site:

  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/cu_runcust_zos.html`
- “Sharing database domains between separate portal instances” is available at the following site:

  `http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.zos.doc/wpf/cu_dbdomnshare_zos.html`

How to get IBM Redbooks

You can search for, view, or download Redbooks®, IBM® Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

WebSphere Portal for z/OS Version 6

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

WebSphere Portal for z/OS Version 6

Creating a WebSphere Portal environment on z/OS

Implementing high availability and security

Configuring and using Web Content Management

A portal is one of the most important components in a Service Oriented Architecture (SOA). IBM WebSphere Portal is available for a variety of platforms, including z/OS and z/Linux.

In this book we discuss IBM WebSphere Portal Enable for z/OS Version 6.0.0.1, which combines the rich functionality brought by IBM WebSphere Portal for Multiplatforms with the Qualities of Service provided by the z/OS platform to provide a robust computing platform.

The information in this book is based on experiences gained during an ITSO Proof of Concept in which we installed and configured WebSphere Portal on z/OS in a highly available and secure environment. It is targeted to those who need to implement Portal in System z environments.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7459-00 ISBN 0738488844