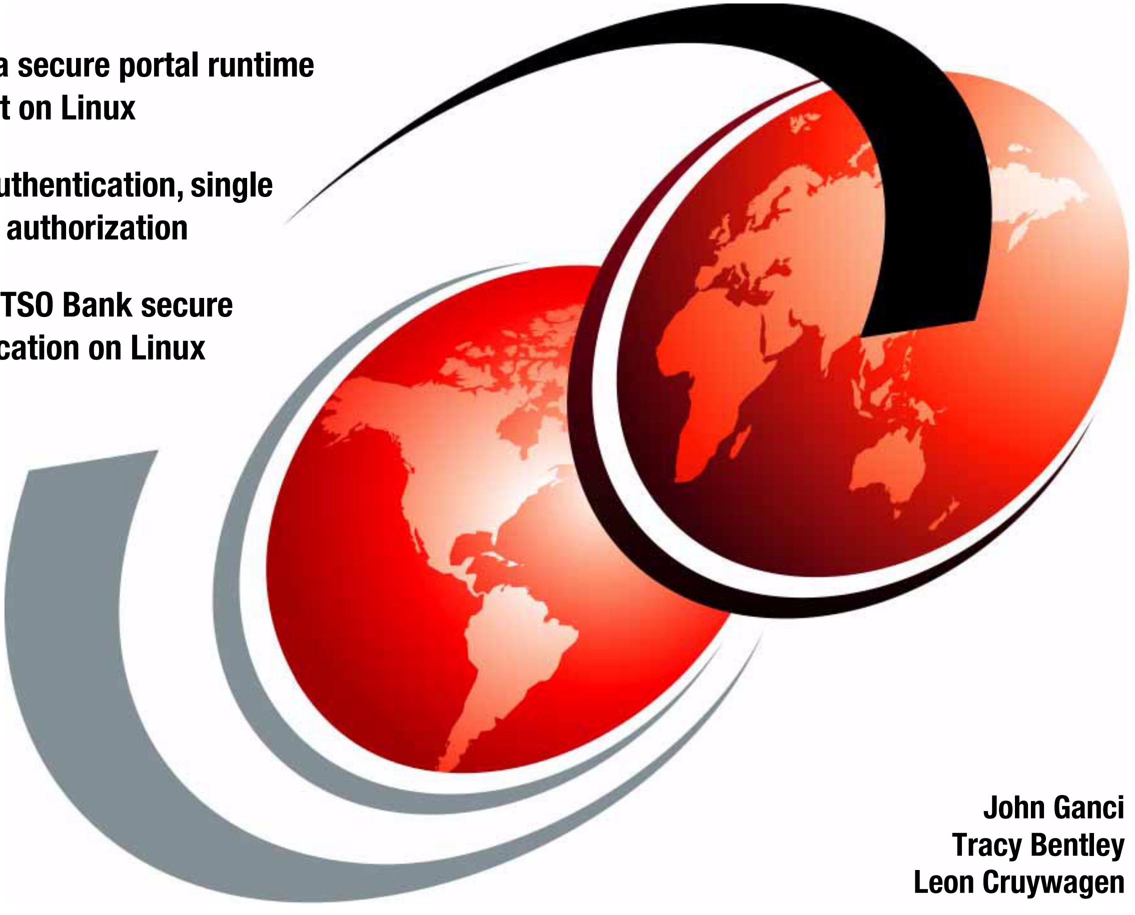
WebSphere® software

IBM

Deploying a Secure Portal Solution on Linux

Using WebSphere Portal V5.0.2 and Tivoli Access Manager V5.1



- Implement a secure portal runtime environment on Linux
- Configure authentication, single signon, and authorization
- Deploy the ITSO Bank secure portal application on Linux

John Ganci
Tracy Bentley
Leon Cruywagen

Redpaper

ibm.com/redbooks

International Technical Support Organization

Deploying a Secure Portal Solution on Linux Using WebSphere Portal V5.0.2 and Tivoli Access Manager V5.1

December 2004

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

First Edition (December 2004)

This edition applies to IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 and IBM Tivoli Access Manager for e-business V5.1.0.4 for Linux on xSeries.

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

eserver®
ibm.com®
xSeries®
AIX®
Cloudscape™
DB2 Universal Database™
DB2®
IBM®
Lotus®
Redbooks™
Redbooks (logo) ™
ThinkPad®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel 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, and service names may be trademarks or service marks of others.

Preface

Portals provide a personalized single point of access to applications, content, and processes through a Web interface. Secure portal solutions are needed to address common security challenges, such as authentication, authorization and single signon.

This IBM Redpaper and the accompanying sample code provide IT architects, IT specialists, and administrators with the critical knowledge to implement the secure portal solution runtime environment and secure an application. The runtime environment includes IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 and IBM Tivoli Access Manager for e-business V5.1.0.4 on the SUSE LINUX Enterprise Server V8 platform.

For more information about the architecture of the secure portal solution and development related topics, we recommend that you refer to the IBM Redbook *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

John Ganci is a Senior Software Engineer, WebSphere® Specialist at the IBM ITSO, Raleigh Center. He writes extensively and teaches classes on WebSphere and related topics. John has 15 years of experience in product and application design, development, system testing, and consulting. His areas of expertise include WebSphere Application Server, e-commerce, portals, pervasive computing, Linux® and Java™ programming.

Tracy Bentley is a Senior IT Specialist from Orlando, FL, working on the Portal Speed Team. She has worked on numerous WebSphere Portal Proof of Concepts (POCs), showing clients how WebSphere Portal will work in their environment. Several of her POCs involved integrating Portal with Tivoli® Access Manager (TAM). Her areas of expertise include runtime configuration and troubleshooting. She also has several years of experience with WebSphere Commerce Suite, and WebSphere MQ. Tracy holds a master’s degree from Webster University.

Leon Cruywagen is a Senior Software Specialist with IBM in the U.S. He has extensive experience in the software design and integration field and is a member of the IBM Professional Certification Board. His areas of expertise include portals, e-commerce and back-end integration with enterprise systems. Leon majored in economics and accounting at the University of Stellenbosch, South Africa.

The IBM Redpaper team from left to right: Leon Cruywagen, Tracy Bentley, John Ganci

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

- Tinny Ng, IBM System House Scenario Designer, Canada
- Kai Schwidder, IBM Consulting Software IT Architect, Switzerland
- Normunds Saumanis, IBM IT Architect, Latvia
- Faheem Altaf, IBM Linux Integration Center, USA

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM® Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction

This redpaper provides detailed procedures to help you implement a secure portal solution on Linux. This chapter begins by looking at the challenges for hosting Web-based applications. Then it presents an overview of the secure portal solution and outlines the software that is involved.

1.1 Challenges for hosting Web-based applications

There are several common challenges for hosting Web-based applications such as a secure portal. First, the site needs to provide a means of determining who is accessing the site (authentication). Second, the site needs the capability to permit or deny access to resources based on the policies and users or groups who access the resources (authorization). Third, users desire to log on only once for access to applications to which they are granted access (single signon (SSO)).

In some cases, businesses have tried to pioneer these solutions on their own. This can be a costly and risky approach to Web-based security. As the complexity of Web sites increases to meet On Demand Business needs, there is a growing expectation for IT shops to deploy solutions in a timely fashion. To solve these infrastructure and security needs, many companies look to leverage middleware software technologies that provide an integrated solution for authentication, authorization and SSO. When companies invest in secure portal solutions from IBM using Tivoli Access Manager and WebSphere Portal, they gain a proven production-ready secure portal solution that can dramatically accelerate their time to market.

For more information about the architecture of the secure portal solution and development-related topics, we recommend that you refer to the IBM Redbook *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

1.2 The secure portal solution

This section presents an overview of the key concepts and the solution architecture of a secure portal solution.

1.2.1 Key concepts of a secure portal solution

There are several key concepts of a secure portal solution when using IBM WebSphere Portal and Tivoli Access Manager. This section briefly defines them.

Authentication

Authentication is a process where the client identity is validated. The client can be an end user, a machine, or an application. Authentication uses the identity of the user, authenticated or unauthenticated, to acquire the credentials of the user. The objective is to determine if the user has the proper permissions for the requested resource.

Authorization

The authorization process provides the capability to permit or deny access to resources based on the policies and users that access the resources. If the resource is protected, the user is first authenticated to determine their identity. Then the privileges that are defined for the desired resource are checked.

Shared LDAP user registry

The user registry is stored under a root Lightweight Directory Access Protocol (LDAP) suffix (for example, dc=itso,dc=ibm,dc=com) in the LDAP repository. In a secure portal solution, Tivoli Access Manager, WebSphere Portal, and WebSphere Application Server reference the same user registry, since they are configured to connect to and use the same Tivoli Directory Server LDAP repository.

Single signon

Single signon provides users with the ability to log on once (authenticate). Then they can access resources or applications within the enterprise to which the user has been granted permissions.

Credential Vault

WebSphere Portal includes the Credential Service and Credential Vault features. They allow portlet applications to pass user credentials to a back-end application. The *Credential Vault* is a portal service that helps portlets and portal users manage multiple identities. When using Tivoli Access Manager with WebSphere Portal to create a secure portal solution, the credential storage for the Credential Vault can be moved to the Tivoli Access Manager Global Signon (GSO) lockbox.

1.2.2 High-level architecture of a secure portal solution

Many possible runtime topologies can be implemented for a secure portal solution, depending on the security, performance, scalability, and integration needs of the business. Figure 1-1 shows the high level secure portal solution architecture. It includes the fictitious ITSO Bank secure portal application. The solution architecture can be applied to many types of applications.

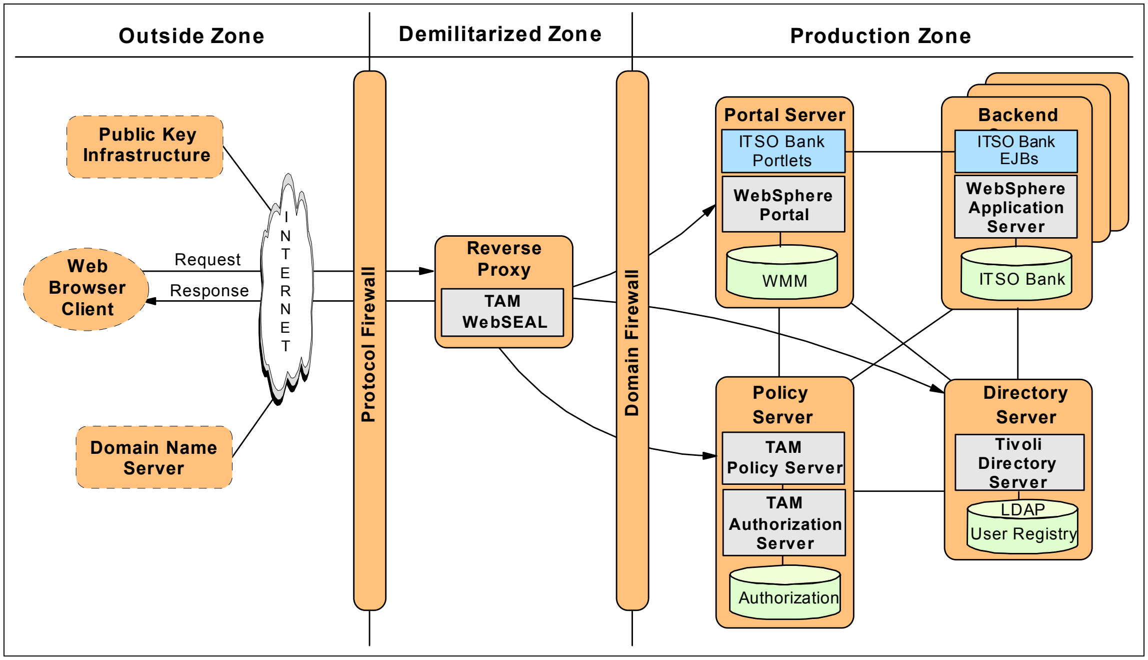


Figure 1-1 High-level architecture of a secure portal solution

The following example illustrates how a customer using a Web browser would interact with the ITSO Bank secure portal solution to access a protected resource such as a customer account balance. First we log on to the ITSO Bank site to outline the process of authentication. Then we highlight the process of authorization to the secure portal page.

1. Authenticate the customer.

   a. The customer enters a URL in the Web browser to access a resource that is protected by WebSEAL.

   b. WebSEAL determines that the user has attempted to access a protected resource and prompts the user with a logon page.

   c. The user enters her username and password in the logon form and then submits them to WebSEAL.

   d. WebSEAL then interacts with the Tivoli Access Manager Policy Server and Tivoli Directory Server to validate the identity of the user in the Tivoli Access Manager user registry.

   e. WebSEAL uses the validated identity to obtain a credential for that user.

2. Authorize access to the secure resource.

   In this example, the client wants to view her account balance.

   a. WebSEAL interacts with the Tivoli Access Manager authorization services and the user credentials to permit or deny access to protected objects (for example, bank account balance) after evaluating the access control list (ACL) permissions and protected object policy (POP).
   b. WebSEAL forwards the request to WebSphere Portal.
   c. The account balance portlet interacts with the back-end Enterprise JavaBeans (EJBs) to retrieve the customer account balance.
   d. WebSEAL sends the response to the Web browser client to display the contents of the portal page.

1.3 Solution software

This section highlights the software that we used in the ITSO working example of a secure portal solution for the runtime environment.

1.3.1 Runtime environment solution software

The majority of the runtime environment software used in the ITSO secure portal solution is included in IBM WebSphere Portal Extend for Multiplatforms V5.0.2 and IBM Tivoli Access Manager for e-business V5.1. In addition, we used the most current fixpack levels of software for these software suites. In some cases, we used them to fix known problems. In other cases, we used them to fully validate the functionality when integrated. We used the SUSE LINUX Enterprise Server V8 with Service Pack 3 as the operating system platform.

Table 1-1 lists the software products and levels included with IBM Tivoli Access Manager for e-business V5.1. It also lists the fixpack levels that we used to implement the secure portal runtime environment for the ITSO working example.

Table 1-1 Software included with Tivoli Access Manager V5.1 and fixpack levels used by the ITSO

Tivoli Access Manager bundled software product name	Tivoli Access Manager bundled software version	ITSO example fixpack version
IBM DB2® Universal Database™ (UDB), Enterprise Server Edition	8.1	8.1.0.56 **Note**: 8.1 + Fixpack 6
IBM GSKit	7.0.1.9	7.0.1.16
IBM Java Runtime Environment (JRE)	1.3.1	1.3.1
IBM WebSphere Application Server **Note**: Used to host Web Administration Tools	5.0.2	5.0.2
IBM Tivoli Directory Server – Directory Server – Directory Client SDK – Web Administration Tool	5.2	5.2.1 **Note**: 5.2 + Fixpack 1
IBM Tivoli Access Manager for e-business – Access Manager Runtime – Access Manager Java Runtime Environment (PDJRTE) – Access Manager Policy Server – Access Manager Authorization Server – Access Manager Web Portal Manager – Access Manager Web Security Environment – Access Manager WebSEAL	5.1	5.1.0.4 **Note**: 5.1 + TAM Base Fixpack 4 + WebSEAL Fixpack 4

Table 1-2 lists the software products and levels included with IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 that we used to implement the secure portal runtime environment for the ITSO working example.

Table 1-2 Software included with WebSphere Portal V5.0.2.2 Extend and fixpack levels used by the ITSO

WebSphere Portal Extend bundled software product name	**WebSphere Portal bundled software version**	**ITSO example fixpack version**
IBM DB2 UDB, Enterprise Server Edition	8.1.1	8.1.0.56 **Note**: 8.1 + Fixpack 6
IBM WebSphere Application Server Enterprise – WebSphere Application Server (Base)	5.0.2.6 **Note**: 5.0 + Fixpack 2 + Cumulative Fix 6	5.0.2.6 **Note**: 5.0 + Fixpack 2 + Cumulative Fix 6
– Programming Module Enhancement (PME)	5.0.2.2 **Note**: 5.0 + PME Fixpack 2 + Cumulative Fix 2	5.0.2.2 **Note**: 5.0 + PME Fixpack 2 + Cumulative Fix 2
IBM Tivoli Directory Server – Directory Server – Directory Client SDK – Web Administration Tool	5.1	5.2
IBM WebSphere Portal Extend for Multiplatforms – WebSphere Portal – WebSphere Portal Content Publisher	5.0.2.2	5.0.2.2

Note: Although we used IBM WebSphere Portal Extend for Multiplatforms, the solutions documented in this redpaper also apply to the Express and Enable Editions.

2

Installing the runtime environment

This chapter explains how to install a secure portal runtime environment on the SUSE LINUX platform for the ITSO working example. The ITSO runtime environment consists of four nodes:

- Reverse Proxy node
- Portal Server node
- Directory Server node
- Policy Server node

The value add of the ITSO working example runtime implementation is three fold. First, we provide detailed procedures for installing the software components by node. Second, we provide sample values for the procedures that put into context the information that is found in the product guides. Third, we include best practices, tips and work-arounds for the procedures based on our first-hand experience.

2.1 Planning

This section describes the scenario for the ITSO working example runtime environment. It also provides information regarding the hardware and software levels we used to implement the environment.

For additional information, refer to the Tivoli Access Manager and WebSphere Portal product guides and the IBM Redbook *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

The implementation procedures for the ITSO working example runtime environment (see Figure 2-1) focus on the following nodes:

- Directory Server node
- Policy Server node
- Reverse Proxy node
- Portal Server node

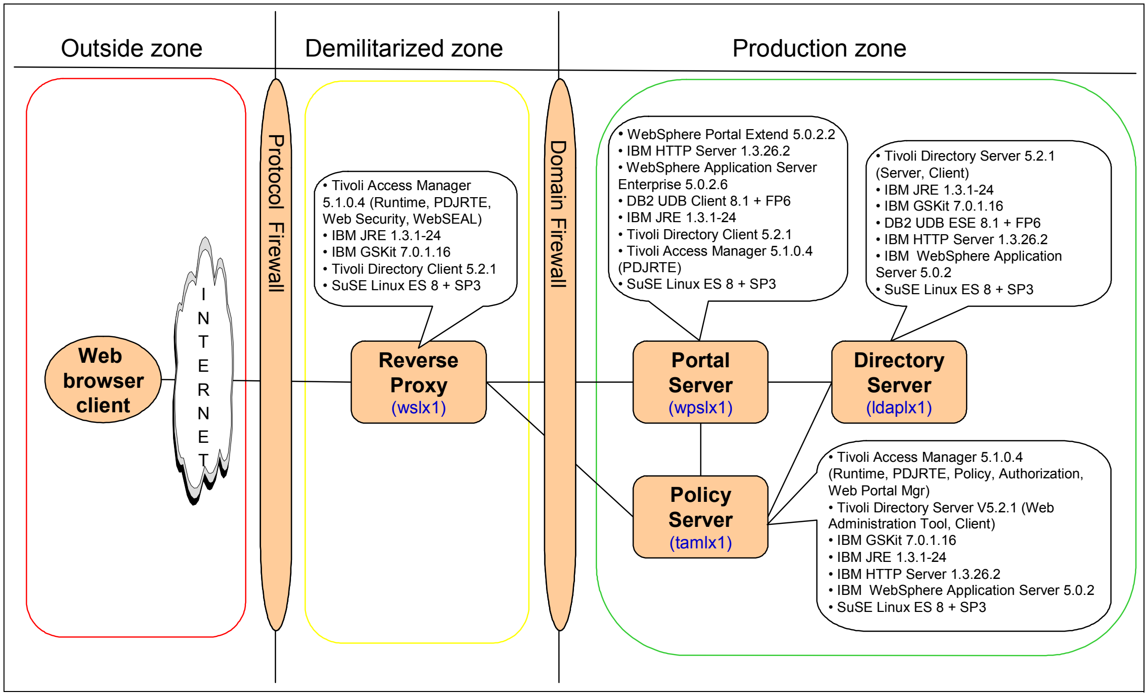

Outside zone
Demilitarized zone
Production zone
Web browser client
INTERNET
Protocol Firewall
• Tivoli Access Manager 5.1.0.4 (Runtime, PDJRTE, Web Security, WebSEAL)
• IBM JRE 1.3.1-24
• IBM GSKit 7.0.1.16
• Tivoli Directory Client 5.2.1
• SuSE Linux ES 8 + SP3
Reverse Proxy (wslx1)
Domain Firewall
• WebSphere Portal Extend 5.0.2.2
• IBM HTTP Server 1.3.26.2
• WebSphere Application Server Enterprise 5.0.2.6
• DB2 UDB Client 8.1 + FP6
• IBM JRE 1.3.1-24
• Tivoli Directory Client 5.2.1
• Tivoli Access Manager 5.1.0.4 (PDJRTE)
• SuSE Linux ES 8 + SP3
• Tivoli Directory Server 5.2.1 (Server, Client)
• IBM JRE 1.3.1-24
• IBM GSKit 7.0.1.16
• DB2 UDB ESE 8.1 + FP6
• IBM HTTP Server 1.3.26.2
• IBM WebSphere Application Server 5.0.2
• SuSE Linux ES 8 + SP3
Portal Server (wpslx1)
Directory Server (ldaplx1)
Policy Server (tamlx1)
• Tivoli Access Manager 5.1.0.4 (Runtime, PDJRTE, Policy, Authorization, Web Portal Mgr)
• Tivoli Directory Server V5.2.1 (Web Administration Tool, Client)
• IBM GSKit 7.0.1.16
• IBM JRE 1.3.1-24
• IBM HTTP Server 1.3.26.2
• IBM WebSphere Application Server 5.0.2
• SuSE Linux ES 8 + SP3

Figure 2-1 ITSO working example: Secure Portal runtime environment on SUSE LINUX

Note: This document does not include procedures for implementing firewalls.

2.1.1 Hardware and software prerequisites

For detailed information about the hardware and software prerequisites, refer to the following product installation guides and Web content:

- *Web Security Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1361
- *Base Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1362
- *Installation and Configuration Guide, IBM Tivoli Directory Server V5.2*, SC32-1338
- IBM WebSphere Portal Extend for Multiplatforms V5.0.2 hardware and software requirements

  `http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html#req5`

2.1.2 Hardware used within the ITSO runtime environment

We used the following hardware within the ITSO working example secure portal runtime environment on SUSE LINUX:

- Directory Server node
  - IBM @server xSeries® 230 (8658-61Y)
    - 1 CPU, Intel® PIII 1 GHz
    - 512 MB main memory
    - 18 GB DASD
    - 1 IBM Ethernet adapter
    - Host name: `ldaplx1.itso.ral.ibm.com`
- Policy Server node
  - xSeries 230 (8658-61Y)
    - 1 CPU, Intel PIII 1 GHz
    - 1024 MB main memory
    - 18 GB DASD
    - 1 IBM Ethernet adapter
    - Host name: `tamlx1.itso.ral.ibm.com`
- Reverse Proxy node
  - xSeries 230 (8658-61Y)
    - 1 CPU, Intel PIII 1 GHz
    - 512 MB main memory
    - 18 GB DASD
    - 1 IBM Ethernet adapter
    - Host name: `wslx1.itso.ral.ibm.com`

- Portal Server node
  - xSeries 230 (8658-61Y)
    - 1 CPU, Intel PIII 1 GHz
    - 1024 MB main memory
    - 18 GB DASD
    - 1 IBM Ethernet adapter
    - Host name: `wpslx1.itso.ral.ibm.com`

Note: For the purpose of prototypes or demos, specialists, and developers may choose to use VMWare Workstation V4.5.2 to run all four nodes on a 2 GB RAM system. For example, it is possible to set up the topology outlined in this redpaper on an IBM ThinkPad® T40 that has 2 GB RAM.

We listed a possible VMWare memory configuration for each of the four nodes that totals to 1.8 GB RAM (leaving roughly 200 MB RAM for the host operating system):

- Directory Server node: 256 MB
- Policy Server node: 512 MB
- Reverse Proxy node: 152 MB
- Portal Server node: 900 MB

After installation and configuration, the nodes can run in text mode, since X11 is not necessary.

2.1.3 Software used within the ITSO runtime environment

We used the software levels in the Table 2-1, Table 2-2, Table 2-3, and Table 2-4 on page 14 within the ITSO working example secure portal runtime environment on SUSE LINUX Enterprise Server V8.

Table 2-1 Directory Server node

Software	Version
SUSE LINUX Enterprise Server	8 + Service Pack 3
IBM Java Runtime Environment (JRE)	1.3.1-24
IBM GSKit	7.0.1.16
IBM DB2 Universal Database (UDB), Enterprise Server Edition	8.1.0.56 **Note**: 8.1 + Fixpack 6a
IBM Tivoli Directory Server – Server – Client SDK	5.2.1 **Note**: 5.2 + Fixpack 1

Table 2-2 Policy Server node

Software	Version
SUSE LINUX Enterprise Server	8 + Service Pack 3
IBM DB2 UDB, Enterprise Server Edition (ESE)	8.1.0.56 **Note**: 8.1 + Fixpack 6a
IBM Java Runtime Environment	1.3.1-24
IBM GSKit	7.0.1.16
IBM WebSphere Application Server	5.0.2 **Note**: 5.0 + Fixpack 2
IBM Tivoli Directory Server – Web Administration Tool – Client SDK (required by TAM Runtime)	5.2.1 **Note**: 5.2 + Fixpack 1
IBM Tivoli Access Manager for e-business – Access Manager Runtime – Access Manager Java Runtime Environment (PDJRTE) – Access Manager Policy Server – Access Manager Authorization Server – Access Manager Web Portal Manager	5.1.0.2 **Note**: 5.1 + Base Fixpack 2

Table 2-3 Reverse Proxy node

Software	Version
SUSE LINUX Enterprise Server	8 + Service Pack 3
IBM GSKit	7.0.1.16
IBM Java Runtime Environment	1.3.1-24
IBM Tivoli Access Manager for e-business – Access Manager Runtime – Access Manager Java Runtime Environment (PDJRTE) – Access Manager Web Security Environment – Access Manager WebSEAL	5.1.0.2 **Note**: 5.1 + WebSEAL Fixpack 2 + Base Fixpack 2

Table 2-4 Portal Server node

Software	Version
SUSE LINUX Enterprise Server	8 + Service Pack 3
IBM WebSphere Application Server Enterprise – WebSphere Application Server (Base) – Programming Module Enhancement (PME)	5.0.2.6 **Note**: 5.0 + Fixpack 2 + Cumulative Base Fix 6 5.0.2.2 **Note**: 5.0 + Fixpack 2 + Cumulative PME Fix 2
IBM WebSphere Portal Extend for Multiplatforms – WebSphere Portal Server – WebSphere Portal Content Publisher	5.0.2.2
IBM DB2 UDB, Client	8.1.0.56 **Note**: 8.1 + Fixpack 6a
IBM Java Runtime Environment	1.3.1-24
IBM Tivoli Access Manager for e-business – Access Manager Java Runtime Environment (PDJRTE)	5.1.0.2 **Note**: 5.1 + Base Fixpack 2

IBM WebSphere Portal Extend for Multiplatforms V5.0.2 CDs

There are 79 CDs included in the IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2. For the ITSO runtime environment on SUSE LINUX Enterprise Server V8, we used the CDs found in Table 2-5.

Note: Due to the vast number IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 CDs and not so obvious naming of the CDs, we provide a listing of the CDs that we used in our environment for reference purposes (Table 2-5).

Table 2-5 IBM WebSphere Portal Extend for Multiplatforms V5.0.2 CDs used for ITSO runtime

CD image directory name	CD description
setup	WebSphere Portal Installer
cd1-3	WebSphere Application Server Enterprise V5.0 for Linux
cd1-8	WebSphere Application Server PTF and Cumulative Fixes for Linux – WebSphere Application Server 5 Base Fixpack 2 (V5.0.2) – WebSphere Application Server 5 PME Cumulative Fix 2 (V5.0.2.2) – WebSphere Application Server 5 Base Cumulative Fix 6 (V5.0.2.6)
cd2	WebSphere Portal Server V5.0.2.2 – Personalization

Software downloads from IBM

In addition to the software included with IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 and IBM Tivoli Access Manager for e-business V5.1, the installation procedures in this chapter provide instructions on how to download and install newer fixpacks and fixes for Tivoli Access Manager than are included in the CD distributions. Also, we use IBM Tivoli Directory Server V5.2 in place of IBM Tivoli Directory Server V5.1 that is included with both WebSphere Portal V5.0.2.2 and Tivoli Access Manager V5.1.

2.1.4 Using VMWare and Ghost

When developing the ITSO runtime implementation procedures on SUSE LINUX, we extensively used VMWare Workstation V4.5.2 so that the Linux operating system could be quickly reloaded to a pristine state. We used VMWare to capture the system state during the installation and configuration of the nodes.

VMWare images can be copied from various system types since the device drivers are virtualized. The downside is that system performance is not as good as on a native operating system.

2.1.5 Unpacking the .taz, .zip, and .tar files to Linux file system

We strongly recommend that you unzip and unpack the .gz or .tar files to a Linux files system (not Microsoft® Windows®). This helps to avoid carriage return line feed characters being inserted into text files (if unpacked to Windows file system). For details about using unpacking utilities supplied with SUSE LINUX, refer to “Common commands” on page 217.

After you unpack the files, we recommend that you change the file permissions. For example, suppose you download the WebSphere Portal CD images to the

/tmp/wp5022 directory and unpack the images to /tmp/wp5022. Change the file permissions as a root user:

```
# cd /tmp
# chmod -R 777 wp5022
```

2.2 Installing the Directory Server node

The Directory Server node is used by WebSphere Portal, Tivoli Access Manager, WebSEAL, and back-end applications as a user repository to facilitate single signon (SSO) and user management. In this example, we install the Tivoli Directory Server Web Administration Tool on the Policy Server node that already has WebSphere Application Server installed to reduce overhead on the Directory Server node.

Important: Consider the following points for Tivoli Directory Server and openldap:

- **openldap server**

  Ensure that the openldap server is not installed.

- **openldap client**

  The openldap-client libraries are installed by default during the SUSE LINUX Enterprise Server V8 installation. The Tivoli Directory Server installer overwrites some of the openldap-client libraries. There are two possible methods to resolve this issue.

  – Uninstall the openldap-client files. Uninstall the openldap client if it is not needed.

  – Move the openldap-client files to avoid configuration conflict and keep shared libraries dependent on the openldap client that is needed by many other Linux packages.

  For the ITSO working example, we chose a manual procedure for coexistence between openldap client and Tivoli Directory Server. For more information, refer to “Tivoli Directory Server and openldap-client” on page 200.

The Directory Server node installation includes the following tasks:

1. Installing SUSE LINUX
2. Directory Server node prerequisites
3. Installing DB2 Universal Database
4. Installing IBM Java Runtime Environment V1.3.1
5. Installing IBM GSKit

6. Installing Tivoli Directory Server V5.2
7. Installing Tivoli Directory Server V5.2 Fixpack 1
8. Configuring Tivoli Directory Server

2.2.1 Installing SUSE LINUX

There are several key issues to address when installing SUSE LINUX Enterprise Server V8 with Service Pack 3. You must also have the latest security updates and patches. For details about installing SUSE LINUX, refer to “Installing SUSE LINUX” on page 200.

2.2.2 Directory Server node prerequisites

Prior to installing the Tivoli Directory Server, ensure that you have completed the following prerequisite steps:

1. Verify that the korn shell is installed. Enter the following command in a terminal `start` window (prerequisite of the DB2 installer):

   ```
   # which ksh
   ```

   This command should return the location of the korn shell as shown in this example:

   ```
   /usr/bin/ksh
   ```

   If you don’t have the korn shell installed, install it before you continue.

2. Verify that you have enough disk space to install the software components on the Directory Server node (refer to Table 2-6). Enter the following command to check the disk space that is available:

   ```
   df -H
   ```

Table 2-6 Directory Server node disk space

Software	Disk space	Target directory
IBM DB2 UDB V8.1 ESE	400 MB	/opt/IBM/db2/V8.1
IBM DB2 UDB V8.1 Fixpack 6a	700 MB	/tmp (the location of tar file)
IBM GSKit V7.0.1.16	24 MB	/opt
IBM Tivoli Directory Server V5.2.1 – Client SDK – Server	156 MB	/usr/ldap

3. Verify that the SUSE LINUX C++ Runtime Libraries are installed.

   During our SUSE LINUX installation, we checked the gcc, gcc-c++, kernel-source, and make. The Tivoli Directory Server V5.2 requires libstdc++-3.2.5. In our example, we have a newer version of this package, since we installed SUSE LINUX V8 Service Pack 3.

   Enter the following command to verify that the SUSE LINUX C++ Runtime Libraries are installed:

   ```
   # rpm -qa | grep libstdc++
   ```

   You should see the following information, provided that you installed SUSE LINUX V8 Service Pack 3:

   ```
   libstdc++-3.3.2-38
   libstdc++-devel-3.2.2-38
   ```

   For details about the installation, refer to “Installing SUSE LINUX Enterprise Server V8” on page 202.

4. Verify that you have the following Perl packages installed on the Directory Server node. These packages are needed for YaST2 to work properly after installing service pack 3, assuming that you installed SUSE LINUX V8 Service Pack 3:

   ```
   perl-XML-DOM-1.39-95.i386
   perl-XML-RegExp-0.03-380.i386
   ```

   To check for these packages, enter the following command:

   ```
   # rpm -qa | grep perl-XML
   ```

   For more information, see “Installing and updating Perl” on page 206.

2.2.3 Installing DB2 Universal Database

This section explains how to install the IBM DB2 Universal Database V8.1, Enterprise Server Edition and supporting Fixpack 6a.

Installing DB2 UDB requires the following tasks:

1. Installing DB2 UDB V8.1
2. Verifying that id db2inst1 is a member of the db2grp1 group
3. Changing the JDK_PATH database parameter
4. Checkpoint: Testing the DB2 UDB server
5. Installing DB2 UDB V8.1 Fixpack 6a
6. Checkpoint: Testing the DB2 UDB server
7. Stopping and starting the DB2 UDB server
8. Creating a test database

Installing DB2 UDB V8.1

To install the IBM DB2 Universal Database V8.1, Enterprise Server Edition, complete the following steps:

> **Note:** Depending on the DB2 UDB V8.1 CD distribution that you are using, the installation panels may be slightly different than the ones that are described here.

1. Open a console window and login as a root user.
2. Insert the *DB2 UDB V8.1 Enterprise Server Edition* CD.
3. Mount the CD-ROM:

   ```
   # mount /dev/cdrom /media/cdrom
   ```

4. Change to the root directory of the CD-ROM:

   ```
   # cd /media/cdrom
   ```

5. Enter the following command to start the DB2 installer:

   ```
   # ./db2setup
   ```

6. In the DB2 Installer window that opens, click **Install Products**.
7. In the Select the Product to Install window, select **DB2 UDB Enterprise Server Edition** and then click **Next**.
8. In the Welcome window for the DB2 Setup Wizard, click **Next.**
9. When you see the License Agreement window, review the agreement and select **Accept** if you are in agreement. Then click **Next**.
10. In the Select the installation type window, select **Typical** and then click **Next**.
11. In the Select the Installation Action window, select **Install DB2 UDB Enterprise Server Edition on this computer** and then click **Next**.
12. In the Set User Information for DB2 Administration Server window, complete the following tasks:

    a. Select the **New user** radio button.
    b. For User name, type `dasusr1`.
    c. For UID, leave this field blank.
    d. For GID, leave this field blank.
    e. For Group name, type `dasadm1`.
    f. Type your password.
    g. Confirm your password.
    h. For Home directory, type `/home/dasusr1`.
    i. Click **Next**.

13. In the Setup a DB2 instance window, select **Create a DB2 instance** and then click **Next**.

14. In the Select how the instance will be used window, select **Single partition instance** (for our scenario), and then click **Next**.

15. In the Select user information for the DB2 instance owner window, complete the following tasks:

    a. Select the **new user** radio button.
    b. For User name, type `db2inst1`.
    c. For UID, leave this field blank.
    d. For Group name, type `db2grp1`.
    e. For GID, leave this field blank.
    f. Type your password.
    g. Confirm your password.
    h. For Home directory, type `/home/db2inst1`.
    i. Click **Next**.

16. In the Select user information for the fenced user window, complete the following tasks:

    a. Select the **new user** radio button.
    b. For User name, type `db2fenc1`.
    c. For UID, leave this field blank.
    d. For Group name, type `db2fgrp1`.
    e. For GID, leave this field blank.
    f. Type your password.
    g. Confirm your password.
    h. For Home directory, type `/home/db2fenc1`.
    i. Click **Next**.

17. In the Prepare the DB2 tools catalog window, select **Do not prepare the DB2 tools catalog on this computer** and then click **Next**.

18. In the Setup Administration contact list, select **Local - Create a contact list on the system**, accept the defaults for the remaining fields, and then click **Next**.

19. If you do not specify an SMTP server, you see a warning message. Click **OK**.

20. In the Specify a contact for health monitor notification window, accept the default option and click **Next**.

21. In the Start copying files window, review the selected options and click **Finish**.

22. When the installation is complete, click the **Status** tab to review the status of the installation. When you are done, click **Finish**.

Verifying that id db2inst1 is a member of the db2grp1 group

After the DB2 UDB installation, verify that id db2inst1 is a member of the db2grp group.

1. To determine if the id db2inst1 user is a member of the db2grp1 group, enter the following command from a console window as root:

   ```
   # id db2inst1
   ```

   The command should return the following result:

   ```
   uid=105(db2inst1) gid=102(db2grp1) groups=102(db2grp1),101(dasadm1)
   ```

2. To list the groups of which db2inst1 is a member, enter:

   ```
   # groups db2inst1
   ```

   The command should return the following result:

   ```
   db2inst1 : db2grp1 dasadm1
   ```

3. To list the groups of which root is a member, enter:

   ```
   # groups root
   ```

   The command should return the following result. While you may have a different list, ensure that db2grp1 is listed.

   ```
   root : root bin pkcs11 db2grp1
   ```

   If db2grp1 is not listed, add root to the group by using the graphical tool for administration or edit the /etc/group file. To edit the /etc/group file, enter:

   ```
   # kate /etc/group
   ```

   The command should open a text editor. Make changes to add root to the group db2grp1:

   ```
   db2grp1:x:102:db2inst1,root
   ```

   Close all existing terminal windows for the desktop to inherit the group changes.

Changing the JDK_PATH database parameter

While installing IBM DB2 UDB V8.1, the JDK_PATH is set to /opt/IBMJava2-131/jre/bin/java. You must change this path to point to /usr/lib/IBMJava2-1.3.1 for certain Java-dependent DB2 functions to work properly, including the database catalog.

1. Switch to the db2inst1user by running the **su -** root command:

   ```
   # su - db2inst1
   ```

2. Update the JDK_PATH by entering the following command:

   ```
   > db2 update dbm cfg using JDK_PATH /usr/lib/IBMJava2-1.3.1
   ```

Checkpoint: Testing the DB2 UDB server

We recommend that you verify that DB2 UDB is working properly prior to installing the fixpack.

1. Log on as a root user.
2. Issue the **xhost** command, to open the xhost server for connections from all clients, while logged in as a root user:

   ```
   # xhost +
   ```

3. Switch to the db2inst1user by running the **su -** root command:

   ```
   # su - db2inst1
   ```

4. Export the display to connect to the X server running on your local system:

   ```
   # export DISPLAY=localhost:0
   ```

5. Echo the display variable to ensure that it is set properly:

   ```
   # echo $DISPLAY
   ```

6. Test that the path has been changed correctly. Type the following command for db2inst1:

   ```
   # db2cc
   ```

   If everything is correct, the DB2 Control Center should load in another window.

> **Note:** If you see an exception that states that you can't connect to the X11 window, then it is likely that you have not set your display variable or access controls (xhost) correctly.

Installing DB2 UDB V8.1 Fixpack 6a

We installed IBM DB2 UDB V8.1 Fixpack 6a for Linux. We chose to use Fixpack 6a for several reasons. First, both Tivoli Directory Server V5.2 and WebSphere Portal V5.0.2.2 officially support DB2 UDB V8.1 Fixpack 6a. Also, we wanted to use the same version of DB2 UDB Fixpack on all nodes for compatibility reasons.

> **Note:** For more information about the contents of the IBM DB2 UDB V8.1 Fixpack 6a, refer to the FixpackReadme.txt file, which you can download from:
>
> ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2linuxIA32v8/fixpak/FP6a_MI00093/

To download and install the IBM DB2 UDB V8.1 Fixpack 6a, follow these steps:

1. Download the IBM DB2 UDB V8.1 Fixpack 6a from the Web at:

   ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2linuxIA32v8/fixpak/FP6a_MI00093/

   We downloaded the FP6a_MI00093.tar file, which is Fixpack 6a for the IBM DB2 Universal Database V8.1, Enterprise Server Edition.

2. Unpack the FP6a_MI00093.tar file to a temporary directory, for example, /tmp/db2v81.fp6.

3. Before you install DB2 Fixpack 6a, ensure that all DB2 processes are stopped.

   a. Switch to root authority by running the **su -** root command.

   b. Run the following commands for each instance:

      ```
      su - iname (iname is the db2 instance name, our example is db2inst1)
      db2 force applications all
      db2 terminate
      db2stop
      db2licd -end      # run at each physical node
      exit
      ```

4. Run the following commands to stop the DB2 Administration Server:

   ```
   su - aname (aname is the administration server, our example is dasusr1)
   db2admin stop
   exit
   ```

5. Log on as a root user.

6. Change to the directory in which the installation image is located, for example, *cd_root*\linux.

7. Enter the **./installFixPak** command to launch the installation as a root user. You should see many DB2 rpms installed.

8. The internal DB2 level should be 8.1.0.56 after you install Fixpack 6a. In our example, we do not have any existing databases that need special attention (rebind DB2 utilities).

   a. Change to the db2inst1 user by entering the following command:

      ```
      # su - db2inst1
      ```

   b. Check the db2 version level to verify that the fixpack was installed successfully:

      ```
      > db2level
      ```

   c. Make sure that the two entries, DB2 v8.1.0.56 and Fixpack 6a, are listed.

9. Update the DB2 instance with the new level of DB2. (This step is *mandatory.*)

   a. Log on as a root user.

   b. Enter the following command for each DB2 instance:

      ```
      # INSTHOME/instance/db2iupdt iname
      ```

      Here *iname* is the instance name and *INSTHOME* is the installation directory. In our example, we enter:

      ```
      # /opt/IBM/db2/V8.1/instance/db2iupdt db2inst1
      ```

10. Update the DB2 Administration Server (DAS) with the new level of DB2 if it is installed.

    a. Log on as a root user.

    b. Enter the following command for each DB2 instance:

       ```
       # INSTHOME/instance/dasuptd dasname
       ```

       Here *dasname* is the DAS owner name and *INSTHOME* is the installation directory. In our example, we enter:

       ```
       # /opt/IBM/db2/v8.1/instance/dasupdt dasusr1
       ```

11. We found that the DB2 server, on occasion, would not start after applying Fixpack 6a and the update procedure. We recommend that you restart the system if this occurs to clear semaphores and memory segments. Or use the **`ipcs`** command to display, and `ipcrm -m <shmid>` to remove, them from memory.

Note: If you created databases before you installed the Fixpack 6a, you must rebind the DB2 utilities to the databases. This step is necessary for the fixes to become effective on an existing database. You must perform the binding procedure only once per database. Note that this is not required for databases that are created after the fixpack is installed. We summarize the rebind procedure found in the FixpackReadme.txt.

To rebind existing DB2 UDB databases after you install Fixpack 6a, enter the following commands from a DB2 command window for each database:

```
db2 terminate
db2 CONNECT TO dbname
db2 BIND DB2_home\BND\@db2ubind.lst BLOCKING ALL GRANT PUBLIC
db2 BIND DB2_home\BND\@db2cli.lst BLOCKING ALL GRANT PUBLIC
db2 terminate
```

Here *dbname* represents the name of a database to which the utilities should be bound. Also *DB2_home* represents the directory where you installed DB2. `db2ubind.lst` and `db2cli.lst` contain lists of required bind files used by IBM DB2 UDB V8.1.

Checkpoint: Testing the DB2 UDB server

After you finish installing DB2 UDB V8.1 Fixpack 6a, we recommend that you verify that DB2 UDB is still working properly. The easiest way to verify this is to run **db2cc**. For details, refer to “Checkpoint: Testing the DB2 UDB server” on page 22.

Stopping and starting the DB2 UDB server

To stop and start the DB2 UDB server, follow these steps:

1. Open a console window as a root user.
2. Log on as the DB2 instance owner:

   ```
   # su - db2inst1
   ```

3. Stop the DB2 UDB server:

   ```
   > db2stop
   ```

4. Start the DB2 UDB server:

   ```
   > db2start
   ```

Creating a test database

We recommend that you create a test database as shown here:

```
# su - db2inst1
> db2 create db sample
```

Later we connect DB2 UDB Client nodes to the DB2 UDB server and verify that we can remotely attach to the sample database.

2.2.4 Installing IBM Java Runtime Environment V1.3.1

For details about configuring the JRE and JAVA_HOME, refer to “Configuring IBM Java Runtime Environment V1.3.1” on page 207.

2.2.5 Installing IBM GSKit

The GSKit is used to manage keystores and certificates. It includes the IBM Key Management utility and libraries accessible to applications to create and manage certificates.

The Tivoli Directory Server V5.2 installs GSKit V7.0.1.9 on the Directory Server node, which includes root certificates that have expired. This results in the administrator not being able to create a new keystore using the iKeyman utility.

In addition, the IBM GSKit V7.0.1.16 addresses a potential denial-of-service attack vulnerability. To avoid the expired certificate problem, we installed the IBM

GSKit V7.0.1.16 before installing Tivoli Directory Server V5.2. This section explains how to download and install IBM GSKit V7.0.1.16.

Determining the GSKit version installed

If you are following the order of installing components documented in this chapter, you can skip this section. It is not necessary since we install the new GSKit prior to installing the components that use it.

Note: To obtain the GSKit version, use the **gsk7ver** command.

If you have not installed Tivoli Directory Server or other software that contains the IBM GSKit, you can skip this section.

To determine the level of the GSKit installed, verify you are logged in as the root authority:

```
# gsk7ver
```

The command should return the following GSKit version.

```
ProductVersion: 7.0.1.16
```

If you have an earlier version, then complete the steps in the following section to install the correct version.

Installing the IBM GSKit V7.0.1.16

Due to updates in the certificates, you need to update the GSKit that is shipped with all the products to the IBM GSKit V7.0.1.16. Since the GSKit includes encryption technology, it is subject to export control. To obtain the updated IBM GSKit V7.0.1.16 and install the GSKit, complete these steps:

1. Obtain IBM GSKit V7.0.1.16 by requesting it from IBM Support at:

   http://techsupport.services.ibm.com/guides/tivoli_contacts.html

2. IBM Support provides a single RPM file that you can install using either an RPM package management tool or from a command line. Ensure that you received the correct file. It must be the following package:

   ```
   gsk7bas-7.0-1.16.i386.rpm
   ```

   Note: Various sets of the IBM GSKits are available, so make sure that you obtain the correct one. If you receive gsk7bas_gcc295-7.0-1.16.rpm, you can make this one work. However, you must relink and copy the files.

3. Copy the GSKit package to a temporary space on your system (for example, /tmp):

   ```
   # cp gsk7bas-7.0-1.16.i386.rpm /tmp
   ```

4. Change to the /tmp directory and issue the following command to install the IBM GSKit:

   ```
   # rpm -Uvh gsk7bas-7.0-1.16.i386.rpm
   ```

   This updates any previous versions of the GSKit if it was previously installed or installs the GSKit if it was not installed.

5. To verify that the GSKit was installed or updated successfully, enter the following command:

   ```
   # rpm -qa | grep gsk
   ```

   This command returns information about the package that is installed. You should see the GSKit package:

   ```
   gsk7bas-7.0-1.16.i386.rpm
   ```

   Or, enter the following command to check the GSKit version:

   ```
   # gsk7ver
   ```

   This command returns the version of all the lib files associated with the GSKit package. Look for the following output:

   ```
   FileVersion: 7.0.1.16
   ```

Configuring JCE and CMS extensions needed for IBM GSKit

This section explains how to set up the IBM GSKit. The IBM GSKit includes the IBM Key Management Utility used to create and manage Certificate Management System (CMS) key database file and certificates.

To configure the JCE extensions for the IBM GSKit, follow these steps:

1. Copy the following files from /usr/local/ibm/gsk7/classes/jre/lib/ext to /usr/lib/IBMJava2-1.3.1/jre/lib/ext:

   ```
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/ibmjceprovider.jar
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/ibmpkcs.jar
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/ibmjcefw.jar
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/local_policy.jar
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/US_export_policy.jar
   /usr/lib/IBMJava2-1.3.1/jre/lib/ext/ibmpkcs11.jar
   ```

2. Change to following directory:

   ```
   # cd /usr/lib/IBMJava2-1.3.1/jre/lib/security
   ```

3. Back up the java.security file (for example, java.security.org).

4. Modify the java.security file. Replace the current providers with the following three providers:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
```

> **Note:** For more information about configuring the IBM CMS service provider, refer to *Setting up the GSKit iKeyman utility* on the Web at:
>
> `http://publib.boulder.ibm.com/tividd/td/ITAMOS/SC32-1362-00/en_US/HTML/am51_install143.htm#gskitikey`

Checkpoint: Testing the IBM Key Management Utility

After the IBM GSKit is configured, we recommend that you verify that the IBM Key Management utility (ikeyman) is working properly.

1. Start the ikeyman utility by entering the following command:

   ```
   # gsk7ikm
   ```

2. From the IBM Key Management utility menu bar, select **Key Database File →New Provider**.
3. Verify that the following entries are in the Provider(s) drop-down list:
   - SUN
   - IBMCMS
   - IBMJCE

2.2.6 Installing Tivoli Directory Server V5.2

This section explains how to install the Tivoli Directory Server V5.2 on the Directory Server node. This section is organized as follows:

- Considerations for installing Tivoli Directory Server
- Moving the openldap-client files
- Creating a DB2 instance owner user and group used by LDAP
- Installing Tivoli Directory Server

Considerations for installing Tivoli Directory Server

Here are some installation considerations for our Tivoli Directory Server environment:

- Both Tivoli Access Manager V5.1 and WebSphere Portal V5.0.2.2 include Tivoli Directory Server V5.1. We used Tivoli Directory Server V5.2 to verify the configuration with the newer version.
- Tivoli Directory Server V5.2 includes WebSphere Application Server Express and the Web Administration Tool used to manage the directory server. We

installed the Web Administration Tool on WebSphere Application Server (not Express), since WebSphere Application Server provides the ability to be configured for security, and Express Edition does not.

- We installed the Web Administration node on the Policy Server node. This is where WebSphere Application Server is installed and will host both the Tivoli Directory Server Web Administration Tool as well as the Tivoli Access Manager Web Portal Manager.
- For the Tivoli Directory Server node we installed only the Directory Server (Server 5.2) and the Directory Server client (Client SDK 5.2).
- To avoid configuration conflicts between the openldap client installed by SUSE LINUX and the Tivoli Directory Server, we moved some of the openldap-client files to a new directory prior to installing the Tivoli Directory Server. We chose not to uninstall the openldap client, since shared libraries that depend on the openldap client being installed are required by many other Linux packages.

Moving the openldap-client files

Many distributions of Linux include the openldap-client. Without making special configuration changes, the openldap client causes problems with the Tivoli Directory Server.

There are a couple of possible solutions to this problem. One option is to uninstall the openldap client. The problem with uninstalling the client is that many supporting libraries are needed by other packages. Another option is to move the openldap-client files to a new directory prior to installing the Tivoli Directory Server.

The following procedure explains how to move the openldap-client files prior to installing the Tivoli Directory Server to avoid configuration conflicts.

1. To determine if the openldap-client is installed, enter the following command from a shell console window:

   ```
   # rpm -qa | grep “ldap.*client”
   ```

   The command should return the following results if openldap-client is installed:

   ```
   openldap2-client-2.1.4-70
   yast2-ldap-client-2.6.5-122
   ```

2. To determine where the openldap-client is installed, enter the following command from a console window:

   ```
   # rpm -q --whatprovides /usr/bin/ldapsearch
   ```

The command should return the following result:

```
openldap2-client-2.1.4-70
```

3. Create a directory to backup the openldap-client files. Enter:

   ```
   # mkdir -p /usr/bin/openldapclient
   ```

4. Change to the /usr/bin directory.
5. Move the following openldap-client files found in the /usr/bin directory to the /usr/bin/openldapclient directory:

   ```
   # mv ldapdelete ldapmodrdn ldapsearch ldapmodify ldapadd
   /usr/bin/openldapclient/
   ```

Creating a DB2 instance owner user and group used by LDAP

This section explains how to create a Linux user and group that will be used as the DB2 instance owner to manage the Tivoli Directory Server LDAP database.

1. Launch YaST2. Click **Start Application** →**System** →**YaST2**.
2. Select **Security and Users** →**Edit and create groups**.
3. Click **Add** to add a new group.
4. On the Add a new group page, complete the following steps:
   a. For Group name, type `ldapgrp`.
   b. For Group id (gid), type `500`.
   c. For Enter a password, leave this field blank.
   d. Re-enter the password, leave this field blank.
   e. In the Members of this group list, select the check box next to **root** user.
   f. Click **Create**.
5. On the User and group administration page, select the **User administration** radio button, and then click **Add** to add a user.
6. On the Add a new user page, complete the following steps:
   a. For First name, type `LDAP`.
   b. For LDAP Name, type `Admin`.
   c. For User login, type `ldapdb2`.
   d. Enter your password.
   e. Re-enter the password for verification.
   f. Click the **Details** button.
7. On the Add/Edit User Properties - Details page, complete the following steps:
   a. For User ID (uid), accept the default.
   b. For Home Directory, type `/home/ldapdb2`.
   c. For Login shell, select **/usr/bin/ksh**.
   d. For Default group, type `ldapgrp`.
   e. Click **Next**.

8. On the Add a new user page, click **Create**.
9. You should now see the ldapdb2 user that you created in the list of users. Click **Finish**.
10. When you see the message indicating that the new user and group settings are configured and usable, click **OK**.
11. Click **Close** to close YaST2.

Installing Tivoli Directory Server

To install the Tivoli Directory Server V5.2 on the Policy Server node, follow these steps:

1. Ensure that you backed up the openldap-client files. For details, refer to "Moving the openldap-client files" on page 29.
2. Insert the *Tivoli Directory Server V5.2* CD.
3. Mount the CD-ROM:

   ```
   # mount /dev/cdrom /media/cdrom
   ```

4. Change to the root directory of the CD-ROM:

   ```
   # cd /media/cdrom
   ```

5. Enter the **`setup`** command to start the Directory installation:

   ```
   # ./setup
   ```

6. Select the installer language (for example English), which is separate from the Tivoli Directory Server language. Click **OK**.
7. In the Welcome panel, click **Next**.
8. When you see the License Agreement, review the terms. If you are in agreement, select **I accepts the terms in the license agreement** and then click **Next**.
9. A panel displays the existing components. In our example, we pre-installed the IBM GSKit 7.0.1.16 and IBM DB2 UDB 8.1.0.56 (both should be displayed). Click **Next**.
10. Select the Tivoli Directory Server language (for example, English), and click **Next**.

11. In the Select the features to install panel (Figure 2-2), select the following options:

   a. Select **Client SDK 5.2**.

   b. Deselect **Web Administration Tool**.

   > **Note**: We install the Tivoli Directory Server Web Administration Tool on the Policy Server node on a WebSphere Application Server shared with the Tivoli Access Manager Web Portal Manager.

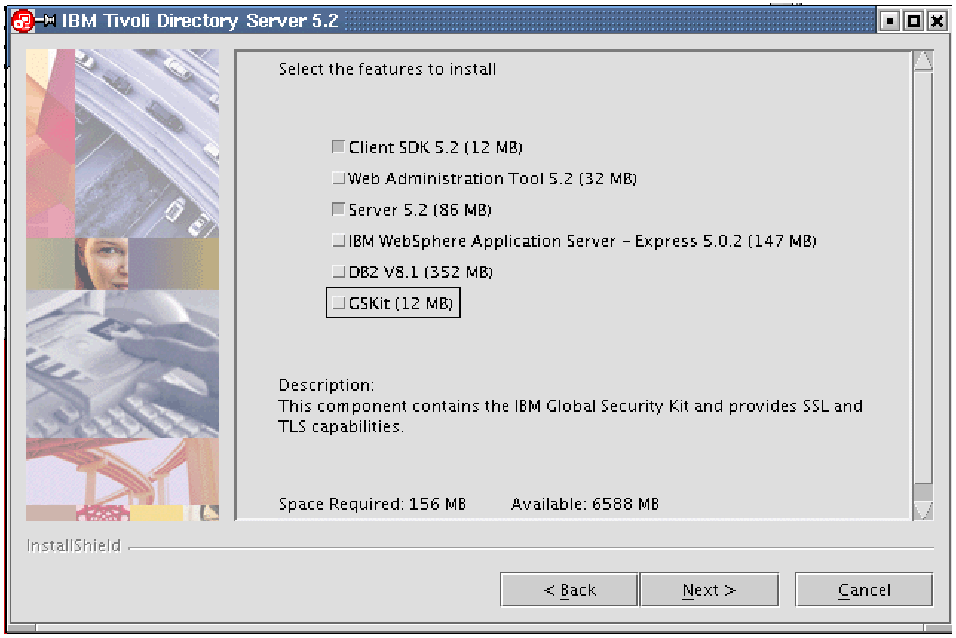
   c. Select **Server 5.2**.

   d. Deselect **IBM WebSphere Application Server - Express 5.0**. We use WebSphere Application Server instead of Express on the Policy Server Node.

   e. Deselect **DB2 V8.1**. A newer level of DB2 UDB is already installed.

   f. Deselect **GSKit**. A new level of the GSKit is already installed.

   g. Click **Next**.



Select the features to install

☑ Client SDK 5.2 (12 MB)
☐ Web Administration Tool 5.2 (32 MB)
☑ Server 5.2 (86 MB)
☐ IBM WebSphere Application Server – Express 5.0.2 (147 MB)
☐ DB2 V8.1 (352 MB)
☐ GSKit (12 MB)

Description:
This component contains the IBM Global Security Kit and provides SSL and TLS capabilities.

Space Required: 156 MB   Available: 6588 MB

Figure 2-2 Tivoli Directory Server: Selecting the features

12. In the Installation Summary panel, review the selections. Then click **Next** to begin installing the files.

13. When the installation is complete, review the readme files for the client and server. Then click **Next**.

14. Click **Finish** and the installer window closes. Then the configuration tool window automatically opens.

2.2.7 Installing Tivoli Directory Server V5.2 Fixpack 1

This section explains how to obtain and install the Tivoli Directory Server V5.2 Fixpack 1. For details about what is included in Fixpack 1, refer to the readme file (FP520-01Readme.pdf), which you can find on the Web at:

```
http://www.ibm.com/support/docview.wss?rs=767&context=SSVJJU&q1=fix+pack&uid=sw
g24007381&loc=en_US&cs=utf-8&lang=en
```

1. Log in as root.
2. Ensure that all of the Tivoli Directory Server client and server processes are stopped prior to installing the Fixpack. Programs and libraries cannot be replaced while they are in use. Note the following items:
   - Tivoli Directory Administration daemon

     For details, refer to “Stopping and starting Tivoli Directory Administration daemon” on page 35.
   - Tivoli Directory Server

     For details, refer to “Stopping and starting the Tivoli Directory Server” on page 36.
   - Web Administration Tool

     In our example, the Web Administration Tool is installed on the Policy Server node. When installing the fixpack on a node where the Web Administration Tool is installed, stop server1 where it was deployed.
   - Client

     Ensure that other applications are not using the Client SDK.
3. Download the Tivoli Directory Server V5.2 Fixpack 1 for x86 Linux (FP520L-01.tar) from the following Web site to a temporary directory (for example, /tmp/tds52.fp1):

   ```
   http://www.ibm.com/support/docview.wss?rs=767&context=SSVJJU&q1=fix+pack&ui
   d=swg24007381&loc=en_US&cs=utf-8&lang=en
   ```
4. Unpack the FP520L-01.tar file to a temporary directory (for example, /tmp/tds52.fp1):

   ```
   # tar -xvf FP520L-01.tar
   ```

5. Start the Tivoli Directory Server V5.2 Fixpack 1 installation by entering the following command from the directory from which you unpacked the fixpack:

   ```
   /tmp/tds52.fp1/FP520L-01/install_update.sh
   ```

   > **Note:** The install_update.sh script saves a backup of the previous version of the Tivoli Directory Server in the /usr/ldap/data directory. Do not remove this directory if you intend to uninstall the update.

6. To verify that Fixpack 1 was installed successfully, confirm that the following file exists:

   ```
   /usr/ldap/bin/FP520-01.txt
   ```

   This file should contain the following information:

   ```
   IBM Directory Release: aus52ldap Build: 040628a
   ```

2.2.8 Configuring Tivoli Directory Server

This section explains how to perform the following base configuration tasks for the Tivoli Directory Server:

- Starting the Tivoli Directory Server Configuration Tool
- Setting the administrator DN and password
- Creating and configuring the directory database
- Stopping and starting Tivoli Directory Administration daemon
- Stopping and starting the Tivoli Directory Server

Starting the Tivoli Directory Server Configuration Tool

To configure Tivoli Directory Server, you must use the Tivoli Directory Server Configuration Tool. If you closed the configuration tool after installation, you can restart the configuration tool by entering the following command:

```
# ldapxcfg
```

Setting the administrator DN and password

To set the administrator distinguished name (DN) and password using the Tivoli Directory Server Configuration Tool, follow these steps:

1. Under Choose a task, select **Administrator DN/password**.
2. Enter the Administrator DN and password:
   a. For Administrator DN, type `cn=root`.
   b. Enter the Administrator Password.
   c. Confirm the password.
   d. Click **OK**.

3. When you see the message `Administrator DN and password successfully updated`, click **OK**.

Creating and configuring the directory database

To create and configure the Tivoli Directory Server directory database from the Tivoli Directory Server - Configuration Tool, follow these steps:

1. Ensure that the DB2 UDB server is started.
2. Under Choose a task, select **Configure database**.
3. Select **Create a new database** and then click **Next**.
4. On the Configure database - user ID page, enter the User ID and password created in “Creating a DB2 instance owner user and group used by LDAP” on page 30 (for example, `ldapdb2`). Then click **Next**.
5. On the Configure database - database name page, enter the database name to be created (for example, `ldapdb`) and click **Next**.
6. On the Configure database - database code page, select **Create a universal DB2 database (UTF-8/UCS-2)** and click **Next**.
7. On the Configure database - database location page, browse to the ldapdb2 home directory (for example, **/home/ldapdb2**) and click **Next**.
8. In the Configuration Summary page, review the selections and click **Finish**.
9. During the configuration, the configuration status is displayed. Notice that a DB2 instance is created with the name of the user designated as the DB2 owner (for example, ldapdb2).

   You should see the following messages if the configuration is successful:

   ```
   Configured IBM Tivoli Directory Server Database.
   IBM Tivoli Directory Server Configuration complete.
   ```

10. When the configuration is complete, click **Close**.

Stopping and starting Tivoli Directory Administration daemon

This section explains how to check the status of the Tivoli Directory Administration daemon, as well as how to stop and start the daemon.

The Tivoli Directory Administration daemon (ibmdiradm) can be found in the directory /usr/ldap/bin.

- Check the status of the Tivoli Directory Administration daemon:

  ```
  # ps -ef | grep ibmdiradm
  ```

  If the daemon is running, you should see a listing of pids for ibmdiradm.

- Stop the Tivoli Directory Administration daemon:
  - If you already configured a directory administration DN and password, you can use the **ibmdirctl** command to stop the administration daemon:

    ```
    # ibmdirictl -D <adminDN> -w <adminPW> admstop
    ```

  - Alternatively, you can end the process ID (pid) from memory:

    ```
    # ps -ef | grep ibmdiradm
    # kill <pid>
    ```

    Here *pid* is returned from the **ps -ef | grep ibmdiradm** command.
- Start the Tivoli Directory Administration daemon:

  ```
  # /usr/ldap/bin/ibmdiradm
  ```

Note: For detailed information about starting and stopping the Tivoli Directory Administration daemon, refer to the *Administration Guide, IBM Tivoli Directory Server V5.2*, SC32-1339, product guide.

Stopping and starting the Tivoli Directory Server

This section explains how to check the status of the Tivoli Directory Server, as well as how to stop and start the server. The **ibmdirctl** command that is used to manage the Tivoli Directory Server requires that the Tivoli Directory Administration daemon (`ibmdiradm`) is started. Also, ensure that the LDAP DB2 instance is started.

ibmdirctl command syntax

```
ibmdirctl [-h hostname] [-D adminDN] [-w <password>] [-p portnumber]
start|stop|restart|status -- [ibmslapd options]
```

- Check the Tivoli Directory Server status:

  ```
  /opt/ldap/bin/ibmdirctl -h ldaplx1 -D cn=root -w password status
  ```

- Stop the Tivoli Directory Server:

  ```
  /opt/ldap/bin/ibmdirctl -h ldaplx1 -D cn=root -w password stop
  ```

- Start the Tivoli Directory Server using either of the following commands:

  ```
  /opt/ldap/bin/ibmdirctl -h ldaplx1 -D cn=root -w password start
  /opt/ldap/bin/ibmslapd
  ```

Note: For detailed information about starting and stopping the Tivoli Directory Server, refer to *Administration Guide, IBM Tivoli Directory Server V5.2*, SC32-1339.

The installation and base configuration for the Directory Server node components are complete.

2.3 Installing the Policy Server node

This section explains how to install and configure the software components on the Policy Server node.

Note: When installing and configuring the Policy Server node, we referenced the following product guides and Redbooks:

- *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325
- *Installation and Configuration Guide, IBM Tivoli Directory Server V5.2*, SC32-1338
- *Administration Guide, IBM Tivoli Directory Server V5.2*, SC32-1339
- *Base Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1360
- *Base Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1362

The high-level tasks to install the Policy Server node are:

1. Installing SUSE LINUX
2. Policy Server node prerequisites
3. Installing WebSphere Application Server
4. Installing Tivoli Directory Server Web Administration Tool
5. Installing IBM Java Runtime Environment V1.3.1
6. Installing the IBM GSKit upgrade
7. Installing Tivoli Directory Client
8. Configuring Directory Server for Tivoli Access Manager
9. Installing Tivoli Access Manager V5.1 base packages
10. Configuring the Tivoli Access Manager base packages
11. Installing Tivoli Access Manager Web Portal Manager
12. Installing Tivoli Access Manager V5.1 Base Fixpack 4
13. Verifying the Tivoli Access Manager servers

2.3.1 Installing SUSE LINUX

For details about installing SUSE LINUX, refer to “Installing SUSE LINUX” on page 200.

2.3.2 Policy Server node prerequisites

Prior to installing the Policy Server node, ensure that you have completed the following steps:

1. Verify that you have enough disk space to install the software components on the Directory Server node (refer to Table 2-7). Enter the following command to check the disk space that is available:

   ```
   df -H
   ```

Table 2-7 Policy Server node disk space

Software	Disk space	Target directory
IBM GSKit V7.0.1.16	10 MB	/usr/local/ibm/gsk
IBM WebSphere Application Server V5.0.2	300 MB	/opt/WebSphere/AppServer
IBM HTTP Server V1.3.26	30 MB	/opt/IBMHttpServer
IBM Tivoli Access Manager V5.1.0.4 – Runtime – Policy Server – Authorization Server – Java Runtime Environment (PDJRTE) – Web Portal Manager	40 MB	/opt/PolicyDirector
IBM Tivoli Directory Server V5.2.1 – Client SDK – Web Administration Tool	150 MB	/usr/ldap

2. Verify that you have the following Perl packages installed on the Directory Server node. These packages are needed for YaST2 to work properly after installing service pack 3 (assuming SUSE LINUX V8 Service Pack 3 is installed):

   ```
   perl-XML-DOM-1.39-95.i386
   perl-XML-RegExp-0.03-380.i386
   ```

   Check for these packages by entering the following command:

   ```
   # rpm -qa | grep perl-XML
   ```

   For details, refer to “Installing and updating Perl” on page 206.

2.3.3 Installing WebSphere Application Server

The WebSphere Application Server on the Policy Server node is used to host the Tivoli Access Manager Web Portal Manager and the Tivoli Directory Server Web Administration Tool.

This section requires the following tasks:

1. Installing WebSphere Application Server V5
2. Verifying WebSphere Application Server V5
3. Installing WebSphere Application Server V5 Fixpack 2 (V5.0.2)

Installing WebSphere Application Server V5

To install WebSphere Application Server V5 on the Policy Server node, complete these steps:

1. Insert the *Tivoli Access Manager V5.1 - Web Administration Interfaces* CD that contains WebSphere Application Server.
2. Mount the CD-ROM.

   ```
   # mount /dev/cdrom /media/cdrom
   ```

3. Change the following directory of the CD-ROM:

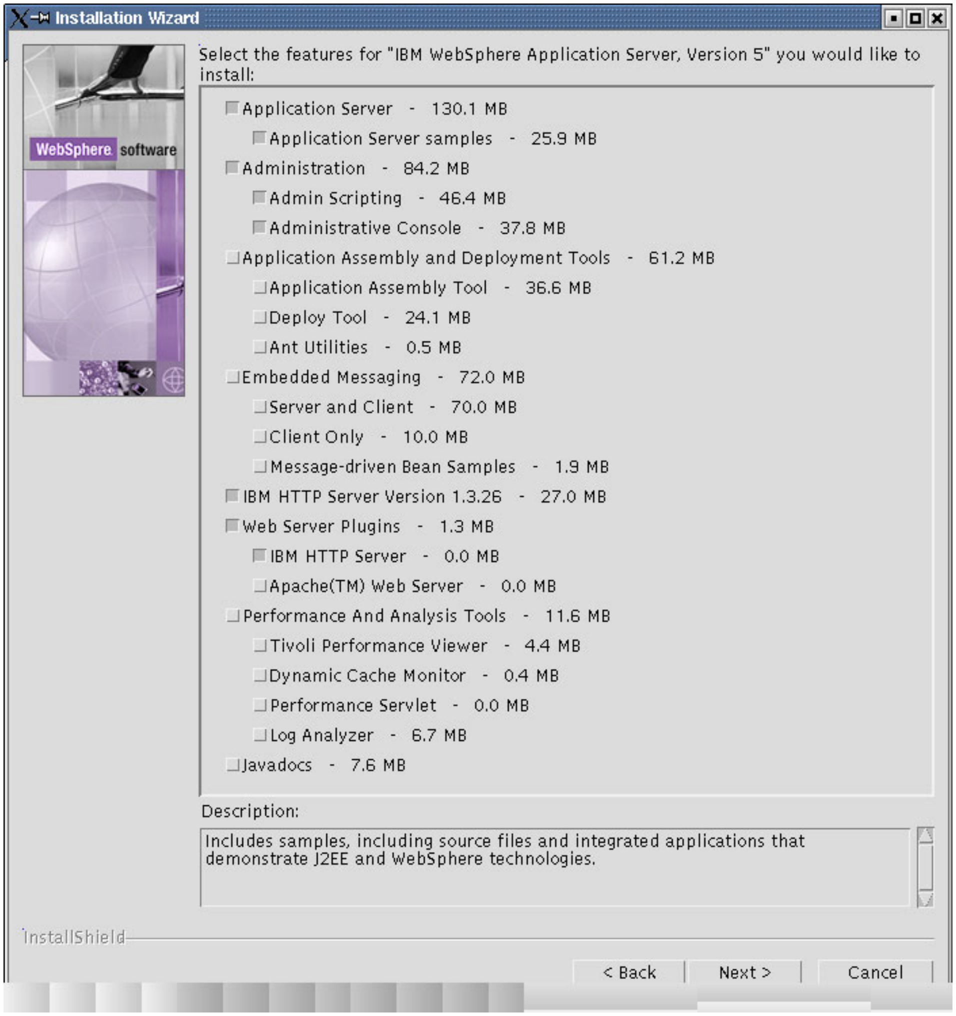
   ```
   # cd /media/cdrom/xSeries/websphere/linui386
   ```

4. To start the WebSphere Application Server Installer, enter:

   ```
   ./install
   ```

5. In the Select the desired language to be used for the installation wizard panel, select the desired language (for example, English) and click **OK**.
6. In the Welcome panel, click **Next**.
7. When you see the License Agreement panel, review the terms and select **I accepts the terms in the license agreement**. Then click **Next**.
8. In the Setup Type panel, select **Custom** and then click **Next**.

9. In the Features Selection panel, select the features. Figure 2-3 shows the features that we selected for this example. Click **Next**.



Select the features for "IBM WebSphere Application Server, Version 5" you would like to install:

- ☑ Application Server - 130.1 MB
  - ☑ Application Server samples - 25.9 MB
- ☑ Administration - 84.2 MB
  - ☑ Admin Scripting - 46.4 MB
  - ☑ Administrative Console - 37.8 MB
- □ Application Assembly and Deployment Tools - 61.2 MB
  - □ Application Assembly Tool - 36.6 MB
  - □ Deploy Tool - 24.1 MB
  - □ Ant Utilities - 0.5 MB
- □ Embedded Messaging - 72.0 MB
  - □ Server and Client - 70.0 MB
  - □ Client Only - 10.0 MB
  - □ Message-driven Bean Samples - 1.9 MB
- ☑ IBM HTTP Server Version 1.3.26 - 27.0 MB
- ☑ Web Server Plugins - 1.3 MB
  - ☑ IBM HTTP Server - 0.0 MB
  - □ Apache(TM) Web Server - 0.0 MB
- □ Performance And Analysis Tools - 11.6 MB
  - □ Tivoli Performance Viewer - 4.4 MB
  - □ Dynamic Cache Monitor - 0.4 MB
  - □ Performance Servlet - 0.0 MB
  - □ Log Analyzer - 6.7 MB
- □ Javadocs - 7.6 MB

Description:

Includes samples, including source files and integrated applications that demonstrate J2EE and WebSphere technologies.

Figure 2-3 WebSphere Application Server selected features

10. In the Features Installation directories panel, specify the directory locations. We entered the following information:

    – WebSphere Application Server: `/opt/WebSphere/AppServer`
    – IBM HTTP Server: `/opt/IBMHttpServer`

    Click **Next**.

11. In the Node name and Host name panel, enter the appropriate names. We entered the following information:

    – Node name: `tamlx1`
    – Host name or IP Address: `tamlx1.itso.ral.ibm.com`

    Click **Next**.

12. In the Installation Summary panel, review your selections. Then click **Next** to begin copying the files.
13. When the installation is complete, you are prompted to register. Click **Next**.
14. First Steps is launched in a separate window. Click **Finish** on the Installation Wizard panel.

Verifying WebSphere Application Server V5

To verify the functionality of the WebSphere Application Server, follow these steps:

1. From the First Steps page, click **Verify Installation**.

    Note: Alternatively, start the First Steps application by entering the following command in a console window as a root user:

    ```
    cd /opt/WebSphere/AppServer/bin
    ./firststeps.sh
    ```

    You should see the `IVT verification succeeded. Installation verification complete.` message.

2. Click **Exit** to close the First Steps application.
3. Ensure that the IBM HTTP Server (WebSphere plug-in) is started. If not, start the server.

    ```
    # cd /opt/IBMHttpServer/bin
    # ./apachectl start
    ```

4. Ensure that the server1 application server is started. If not, start the server as follows:

    ```
    # cd /opt/WebSphere/AppServer/bin
    # ./startServer.sh server1
    ```

    Note: Review the status of the server startup in the startServer.log. For example, we used the GNU utility to view the logs:

    ```
    tail -f /opt/WebSphere/AppServer/logs/server1/startServer.logs
    ```

    You see the `Server server1 open for e-business` message. The server1 directory is not created until the first time the application server is started.

5. Start the WebSphere Application Server Administration Console, by entering the following URL in a Web browser:

   ```
   http://was_hostname:9090/admin
   ```

6. Log on to the WebSphere Administration Console (for example, `admin`).
7. Verify the WebSphere Administration Console. Then click **Logout** to close the Web browser.
8. Stop the server1 application server. Enter:

   ```
   cd /opt/WebSphere/AppServer/bin
   ./stopServer.sh server1
   ```

Installing WebSphere Application Server V5 Fixpack 2 (V5.0.2)

To install the IBM WebSphere Application Server V5 Fixpack 2, follow these steps:

1. Stop WebSphere Application Server, and all application servers and nodes:

   ```
   # /opt/WebSphere/AppServer/bin/stopServer.sh server1
   ```

2. Stop the IBM HTTP Server and IBM HTTP Administration Windows service (WebSphere plug-in fixes):

   ```
   # /opt/IBMHttpServer/bin/apachectl stop
   ```

   > **Note:** The Fixpack attempts to update the IBM HTTP Server and is not able to update the server if it is started.

3. Obtain WebSphere Application Server V5 Fixpack 2 using either of the following methods.

   – Download WebSphere Application Server V5 Fixpack 2 at:

     http://www.ibm.com/support/docview.wss?rs=860&context=SW600&q1=fixpack+2&uid=swg24005012&loc=en_US&cs=utf-8&lang=en+en

   – Use the Tivoli Access Manager V5.1 - WebSphere Fixpack CD, which contains WebSphere Application Server V5 Fixpack 2.

4. Unpack the WebSphere Application Server V5 Fixpack 2 to a temporary directory on the target system (for example, /tmp/was5.fp2).

   > **Note:** The WebSphere Update Installer Wizard needs write access.

5. From the command window, start the WebSphere Installation Update Wizard by entering **`updateWizard.sh`** found in the temporary directory (for example, `/tmp/was5.fp2`).

6. In the WebSphere Update Installer language window, select the appropriate language for the wizard (for example, English) and then click **OK**.
7. In the Welcome window, click **Next**.
8. The WebSphere Update Installer should detect your current WebSphere Application Server version and installation directory (for example, /opt/WebSphere/AppServer). Click **Next**.
9. Select **Install fix packs** and then click **Next**.
10. Enter the directory where you have copied the fixpack. For example, we entered `/tmp/was5.fp2/fixpacks` in the Fix pack directory text field. Then click **Next**.
11. Select the **was50_fp2_linux** fixpack (default) and then click **Next**.
12. You are prompted for the directories for the IBM HTTP Server and the WebSphere Application Server Embedded Messaging (not installed). In this example, we entered the following information:
    a. Select **IBM HTTP Server**.
    b. For IBM HTTP Server installation directory, specify `/opt/IBMHttpServer`.
    c. Deselect **Embedded Messaging (not installed)**.
    d. Click **Next**.
13. Review the fixpack settings. Then click **Next** to begin the fixpack installation of files.
14. When the WebSphere Application Server V5 Fixpack 2 installation is complete, click **Finish**.
15. Verify the WebSphere Application Server after the installation of the fixpack. For details, refer to “Verifying WebSphere Application Server V5” on page 41.

2.3.4 Installing Tivoli Directory Server Web Administration Tool

For the ITSO working example, we chose to install the Web Administration Tool on WebSphere Application Server V5.0.2.6, which is capable of being configured for WebSphere Application Server security.

This section is requires the following tasks:

1. Installing the Installing the Mozilla Web browser plug-in
2. Installing Web Administration Tool
3. Deploying the Web Administration Tool
4. Configuring the Web Administration Tool
5. Verifying the Tivoli Directory Server

Installing the Mozilla Web browser plug-in

If you plan to use the Mozilla Web browser from a SUSE LINUX system to access the Tivoli Directory Server Web Administration Tool, you need to install a plug-in. For details, refer to “Installing Mozilla Web browser (optional)” on page 210.

Installing Web Administration Tool

For details about installing Tivoli Directory Server V5.2 components, refer to “Installing Tivoli Directory Server” on page 31. In this case, we install only the Web Administration Tool 5.2 component.

Deploying the Web Administration Tool

To deploy the Web Administration Tool on the WebSphere Application Server server1 on the Policy Server node, complete these steps:

1. Ensure the Tivoli Directory Server is started on the Directory Server node. For details, refer to “Stopping and starting the Tivoli Directory Server” on page 36.
2. Ensure that the WebSphere Application Server - server1 is started. If not, start server1 as follows:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./startServer.sh server1
   ```

3. Start WebSphere Application Server Administration Console by entering the following URL in a Web browser:

   `http://`*`hostname`*`:9090/admin`

4. Log on to the WebSphere Administration Console (for example, `admin`).
5. In the console navigation tree, click **Applications** →**Install New Application**.
6. In the Preparing for application installation panel, complete these tasks:
   a. For Path, select the **Local path** radio button.
   b. For Local Path, type `/usr/ldap/idstools/IDSWebApp.war`.

      This is the full path of the Web Administration Tool application stand-alone IDSWebApp.war file.

   c. For Context Root, type `/IDSWebApp`.
   d. Click **Next**.
7. In the Generate bindings panel, accept the default settings and click **Next**.
8. In the Step 1: Provide options to perform the installation panel, accept the default settings for Application Name as I`DSWebApp_war` (default). Click **Next**.

9. In the Step 2: Map virtual hosts for Web modules panel, specify the virtual host and Web module. We specified the following options:

   - For Virtual Host, select **default_host** (default).
   - For Web Module, select **IBM Tivoli Directory Server Web Application v2.0**.

   Click **Next**.

10. In the Step 3: Map modules to application servers panel, accept the default and click **Next**.

11. In the Step 4: Summary Review installation options panel, click **Finish**.

    You should see the `Application IDSWebApp_war installed successfully.` message.

12. When the configuration update is complete, click the **Save to Master Configuration** link.

13. When the Save to Master Configuration page opens, click **Save**.

14. .From the WebSphere Application Server Console, complete these tasks:

    a. From the Web Administration Tool, click **Applications** →**Enterprise Applications**.

    b. From the Enterprise Application page, select **IDSWebApp_war** and then click **Start**.

15. Click **Logout** to log out of the WebSphere Application Server Administrative Console.

Configuring the Web Administration Tool

This section explains how to configure the Tivoli Directory Server Web Administration Tool.

Defining the directory server node to the Web Administration Tool

Define the directory server with the Web Administration Tool using these steps:

1. Ensure that the WebSphere Application Server server1 is started.

2. Access the Web Administration Tool from a Web browser:

   ```
   http://tam_hostname:9080/IDSWebApp/IDSjsp/Login.jsp
   ```

   Here *tam_hostname* is the host name where the Tivoli Directory Server Web Administration Tool is deployed.

3. From the Web Administration Tool, complete the following steps:

   a. From the LDAP Hostname drop-down list, select **Console Admin**.
   b. For Username, type `superadmin` (default).

   c. For Password, type `secret` (default).
   d. Click **Login**.

4. Modify the default Console Administration user ID and password.

   a. Select **Console Administration** →**Change console administration login**.

   b. When the Change Console administrator logon opens, complete the following tasks:

      i. For Console administrator login, type `webadmin`.

         We created the administrator webadmin, but this can be any name you desire.

      ii. For Current password, type your password. The default is secret.
      iii. Click **OK.**

   c. Select **Change console administrator password**. Enter the current and new password. Click **OK**.

5. Add the Directory Server node.

   a. Click **Console administration** →**Manage console servers**.

   b. Click **Add**.

   c. Enter the Directory Server host name and change the port numbers if you are not using the defaults (for example, `ldaplx1.itso.ral.ibm.com`):

      i. For Host name, type `ldaplx1.itso.ral.ibm.com`.
      ii. For Port, type `389`.
      iii. For Administration port, type `3538`.
      iv. Deselect **SSL enabled** (default).
      v. Click **OK**.

      You should see the new server listed after you add it.

   d. Click **Close**.

   e. Click **Logout** from the Web Administration Tool.

Changing the password encryption method

We recommend that you change the password encryption method from the default *imask* to SHA or crypt from the Web Administration Tool. The primary reason is that imask is a two-way encoding format and both SHA and crypt have a one-way format.

To configure the password encryption, we used these steps:

1. From the Web Administration Tool, complete the following tasks:
   a. From the drop-down list on the Login page, select the newly created server (for example, `ldaplx1.itso.ral.ibm.com`).
   b. For Username, type `cn=root`.
   c. Enter the password.
   d. Click **Login.**
2. From the Web Administration Tool, select **Server administration** →**Manage security properties.**
3. From the Manage security properties window, click **Password policy**.
4. From the Password encryption drop-down list, select **SHA** and then click **OK** at the bottom of the page.
5. Click **Logout**.

Verifying the Tivoli Directory Server

Now that the Web Administration Tool is configured for the directory server, we recommend that you verify that it is working properly. You do this by connecting to the directory server.

1. Ensure that the Tivoli Directory Server V5.2 Windows service is started.
2. Access the Tivoli Directory Server Web Administration Tool from a Web browser:

   `http://`*`tam_hostname`*`:9080/IDSWebApp/IDSjsp/Login.jsp`
3. From the Web Administration Tool, complete the following tasks:
   a. Select the newly created server (for example, **ldaplx1.itso.ral.ibm.com**) from the drop-down list on the Login page.
   b. For Username, type `cn=root`.
   c. Enter the password.
   d. Click **Login**.
4. Start the Tivoli Directory Server. Click **Server administration** → **Start/stop/restart server**.
5. Click **Start** (do not Check Start/restart in configuration only mode).

   You should see the `Server started` status message.

2.3.5 Installing IBM Java Runtime Environment V1.3.1

For details about configuring the JRE and JAVA_HOME, refer to “Configuring IBM Java Runtime Environment V1.3.1” on page 207.

2.3.6 Installing the IBM GSKit upgrade

For details about installing the IBM GSKit, refer to 2.2.5, “Installing IBM GSKit” on page 25.

2.3.7 Installing Tivoli Directory Client

As a prerequisite to the Tivoli Access Manager Runtime, you must install the Tivoli Directory Server - Client Software Developer Kit (SDK).

1. Move the existing openldap-client files. For details, see “Moving the openldap-client files” on page 29.
2. Install the Tivoli Directory Client.
   a. Install the Tivoli Directory Client V5.2. However select only the **Client SDK** option. For details, refer to 2.2.6, “Installing Tivoli Directory Server V5.2” on page 28.
   b. Install the Tivoli Directory Server V5.2 Fixpack 1. For details, refer to 2.2.7, “Installing Tivoli Directory Server V5.2 Fixpack 1” on page 33.

2.3.8 Configuring Directory Server for Tivoli Access Manager

This section describes the configuration that is required between the Tivoli Directory Server and Tivoli Access Manager.

Preparing the schema definitions

The Tivoli Access Manager schema definitions are added automatically during the installation of the Tivoli Directory Server V5.2.

> **Note:** If you are using IBM Directory Server V4.1 or V5.1, refer to the *Base Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1362, for additional steps for preparing the schema.

Creating a suffix for Tivoli Access Manager metadata

To create a suffix from the Tivoli Directory Server - Web Administration Tool, to be used for Tivoli Access Manager metadata, follow this sequence:

1. Ensure that the Tivoli Directory Server V5.2 Windows service is started.
2. Access the Tivoli Directory Server Web Administration Tool from a Web browser:

   ```
   http://tam_hostname:9080/IDSWebApp/IDSjsp/Login.jsp
   ```

3. From the Web Administration Tool, complete these steps:
   a. From the drop-down list on the Login page, select the newly created server (for example, `ldaplx1.itso.ral.ibm.com`).
   b. For Username, type `cn=root`.
   c. Enter a password.
   d. Click **Login**.
4. Select **Server administration** →**Manager server properties**.
5. On the Manage server properties page, complete these tasks:
   a. Click **Suffixes**.
   b. For the Suffix DN, type `secAuthority=Default`.
   c. Click **Add**.
   d. Click **OK** at the bottom of the page to save the settings.

   Attention: The first time we configured the runtime environment, later while troubleshooting, we discovered that the suffix was not saved. On our system, the display resolution was set to 1024x768. Therefore, the OK button to save the settings was not visible unless we scrolled down the page. Therefore, you may need to scroll down to locate the OK button and click it.

6. Click **Logout**, and close the Tivoli Directory Server - Web Administration Tool.

2.3.9 Installing Tivoli Access Manager V5.1 base packages

This section explains how to install the Tivoli Access Manager V5.1 on the Policy Server node for the ITSO example. When installing Tivoli Access Manager V5.1, you can set up the system using one of the following installation methods:

- Installation using native installation utilities by platform

  The native installer provides for greater control of how and where the components are installed.

- Installation wizard

  This method is useful if you are not experienced with using Tivoli Access Manager and want to quickly implement a base configuration.

To install the Tivoli Access Manager base packages using the native rpm installer, complete the following steps:

1. Insert the *Tivoli Access Manager Base for Linux* CD.
2. Open a console window as a root user.

3. Mount the CD-ROM:

```
# mount /dev/cdrom /media/cdrom
```

4. Change to the directory of the Tivoli Access Manager rpms found on the CD:

```
# cd /media/cdrom/xSeries
```

5. Install the Tivoli Access Manager rpms. Enter:

```
# rpm -ivh PDRTE-PD-5.1.0-0.i386.rpm PDMgr-PD-5.1.0-0.i386.rpm
PDAcld-PD-5.1.0-0.i386.rpm PDJrte-PD-5.1.0-0.i386.rpm
```

   The rpms that we installed are defined in Table 2-8.

Table 2-8 Tivoli Access Manager V5.1 rpms

Tivoli Access Manager rpm	Description
PDRTE-PD-5.1.0-0.i386.rpm	Access Manager Runtime
PDMgr-PD-5.1.0-0.i386.rpm	Access Manager Policy Server
PDAcld-PD-5.1.0-0.i386.rpm	Access Manager Authorization Server
PDJrte-PD-5.1.0-0.i386.rpm	Access Manager Java Runtime Environment

6. Unmount the CD-ROM:

```
# cd /
# umount /media/cdrom
```

2.3.10 Installing Tivoli Access Manager V5.1 Base Fixpack 4

To install Tivoli Access Manager V5.1 Base Fixpack 4, follow these steps:

> **Note:** For detailed information about installing the Tivoli Access Manager V5.1 Base Fixpack 4, refer to the readme file, which you can find at:
>
> http://www.ibm.com/support/docview.wss?rs=638&context=SSPREK&uid=swg24007368&loc=en_US&cs=utf-8&lang=en

1. Ensure that you have installed IBM GSKit V7.0.1.16, which is a prerequisite to Fixpack 4.

   Refer to 2.2.5, “Installing IBM GSKit” on page 25, for details.

2. Ensure that the Tivoli Access Manager services are stopped before you install the fixpack:

   – Access Manager Authorization Server
   – Access Manager Policy Server

The following command stops both servers:

```
# cd /opt/PolicyDirector/bin
# ./pd_start stop
```

3. Stop the server1 application server, where Web Portal Manager has been deployed:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./stopServer.sh server1
   ```

4. Stop the IBM HTTP Server:

   ```
   # cd /opt/IBMHTTPServer/bin
   # ./apachectl stop
   ```

5. Download Tivoli Access Manager V5.1 Base Fixpack 4 (5.1-TAM-FP04_LIN.tar) from the following Web site to a temporary directory (for example, /tmp/tam51-base.fp4) on the Policy Server node:

   http://www.ibm.com/support/docview.wss?rs=638&context=SSPREK&uid=swg24007368&loc=en_US&cs=utf-8&lang=en

6. Open a console window as root.
7. List the Tivoli Access Manager packages that are currently installed:

   ```
   # rpm -qa | grep PD
   ```

   Verify that the following packages are listed:

   ```
   PDMgr-PD-5.1.0-0
   PDAcld-PD-5.1.0-0
   PDRTE-PD-5.1.0-0
   PDJrte-PD-5.1.0-0
   PDWPM-PD-5.1.0-0
   ```

8. Change to the temporary directory where you downloaded the fixpack (for example, `/tmp/tam51-base.fp4`).
9. Unpack the fixpack as follows from the temporary directory (for example, `/tmp/tam51-base.fp4`):

   ```
   # tar -xvf 5.1-TAM-FP04_LIN.tar
   ```

10. Run the following command on one line to update each package to the 5.1.0.4 level:

    ```
    # rpm -Uvh PDMgr-PD-5.1.0-4.i386.rpm PDAcld-PD-5.1.0-4.i386.rpm
    PDRTE-PD-5.1.0-4.i386.rpm PDJrte-PD-5.1.0-4.i386.rpm
    PDWPM-PD-5.1.0-4.i386.rpm
    ```

    > **Note:** If Tivoli Access Manager is already configured, you may need to install it with the **`--noscripts`** flag:
    >
    > `rpm -U --noscripts` *`patchname`*

11. Verify that the packages were updated correctly:

```
# rpm -qa | grep PD
```

Verify that the packages were updated to 5.1.0-4:

```
PDAcld-PD-5.1.0-4
PDMgr-PD-5.1.0-4
PDJrte-PD-5.1.0-4
PDRTE-PD-5.1.0-4
PDWPM-PD-5.1.0-4
```

2.3.11 Configuring the Tivoli Access Manager base packages

Now that the Tivoli Access Manager base packages are installed, you must configure them. We configure the following Tivoli Access Manager base packages using the **pdconfig** utility:

- Access Manager Runtime
- Access Manager Policy Server
- Access Manager Authorization Server
- Access Manager Java Runtime Environment (PDJRTE)

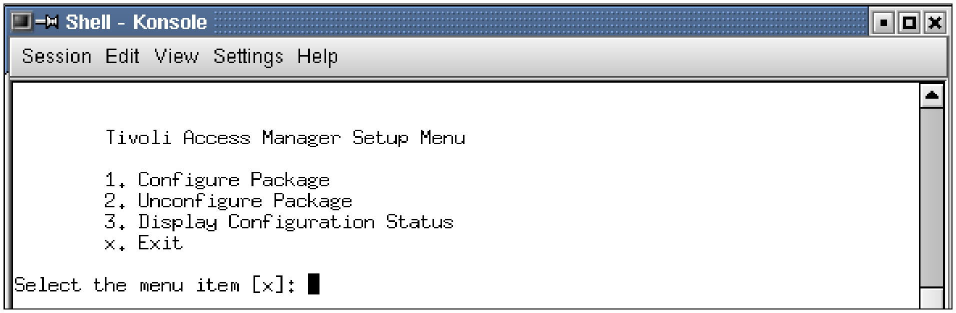
Configuring the Access Manager Runtime

To configure the Tivoli Access Manager Runtime using the **pdconfig** utility, complete these steps:

1. Open a console window as a root user.
2. Start the **pdconfig** utility:

```
# pdconfig
```

3. In the Tivoli Access Manager Setup Menu (Figure 2-4), type 1 (Configure Package) and press Enter.



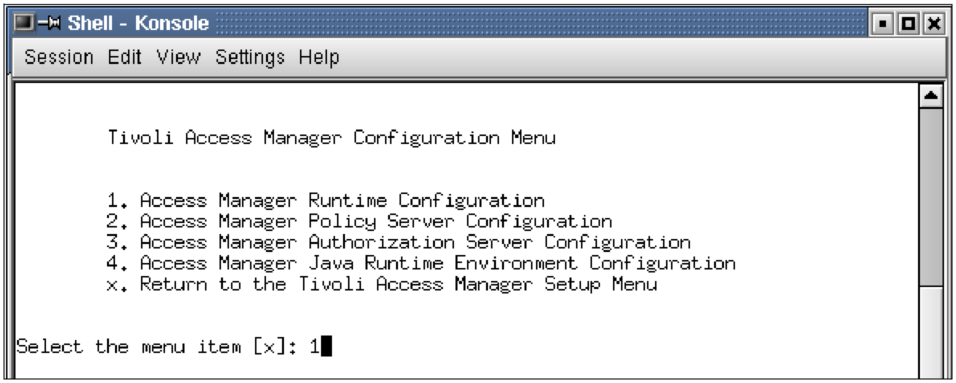
```
        Tivoli Access Manager Setup Menu

        1. Configure Package
        2. Unconfigure Package
        3. Display Configuration Status
        x. Exit

Select the menu item [x]:
```

Figure 2-4 Tivoli Access Manager Setup Menu

4. In the Tivoli Access Manager Configuration Menu (Figure 2-5), type `1` (Access Manager Runtime Configuration) and press Enter.



```
        Tivoli Access Manager Configuration Menu


        1. Access Manager Runtime Configuration
        2. Access Manager Policy Server Configuration
        3. Access Manager Authorization Server Configuration
        4. Access Manager Java Runtime Environment Configuration
        x. Return to the Tivoli Access Manager Setup Menu


Select the menu item [x]: 1
```

Figure 2-5 Tivoli Access Manager Configuration Menu

5. When prompted to configure Tivoli Common Directory logging, press Enter to accept the default of not using Tivoli Common Directory logging.
6. You are prompted with the message `Log file for this application is created in the directory: /var/PolicyDirector/log directory`. Press Enter to accept the default Registry type LDAP.
7. Enter the LDAP server host name. We entered `ldaplx1.itso.ral.ibm.com`. Then press Enter.
8. Enter the LDAP server port. We accepted the default port (389). Then press Enter.
9. Press Enter to return to the configuration menu.

Configuring the Access Manager Policy Server

To configure the Tivoli Access Manager Policy Server using the **`pdconfig`** utility, use a similar process as in “Configuring the Access Manager Runtime” on page 52.

We provide the ITSO working example sample values used to configure the Tivoli Access Manager Policy Server:

- LDAP administrator ID: `cn=root`
- LDAP administrator password: `<password>`
- SSL between Policy Server and LDAP: `no`
- TAM administrator password: `<password>`
- Policy server SSL port: `7135`
- SSL certificate life cycle: `365`

Configuring the Access Manager Authorization Server

To configure the Tivoli Access Manager Authorization Server using the **pdconfig** utility, use a similar process as in “Configuring the Access Manager Runtime” on page 52.

We provide the ITSO working example sample values used to configure the Tivoli Access Manager Authorization Server:

- SSL between Policy Server and LDAP: `no`
- Domain: `Default`
- Policy Server host name: `tamlx1.itso.ral.ibm.com`
- Policy Server SSL port: `7135`
- Administrator ID: `sec_master`
- Administrator password: `<password>`
- Local host name: `tamlx1.itso.ral.ibm.com`
- Administration request port: `7137`
- Authorization request port: `7136`

Configuring the Access Manager Java Runtime Environment

To configure the Tivoli Access Manager Java Runtime Environment (PDJRTE) using the **pdconfig** utility, use a similar process as in “Configuring the Access Manager Runtime” on page 52.

We provide the ITSO working example sample values used to configure the Tivoli Access Manager Java Runtime Environment (PDJRTE):

- Full path of the JRE: `/opt/WebSphere/AppServer/java/jre`
- Full or standalone: `Full`
- Policy Server host name: `tamlx1.itso.ral.ibm.com`
- Policy Server SSL port: `7135`
- Policy Server Domain: `Default`
- Tivoli Common Directory Logging: `No`

> **Note:** After you click Next, you may receive a Data validation error message that states `HPDHP0203E`. If you receive this error message, specify a JRE path that does not contain /lib/ext/ibmjcaprovider.jar or stop this process and then remove the /usr/lib/java/jre/lib/ext/ibmjcaprovider.jar file. The ibmjcaprovider.jar file is depreciated, so we renamed the ibmjcaprovider.jar file to `ibmjcaprovider.old` to avoid this error.

Verifying that all Tivoli Access Manager base packages are configured

From the **pdconfig** utility Tivoli Access Manager Setup Menu, type `3` for Display Configuration Status. You should see that the status of all base packages are *configured*.

2.3.12 Installing Tivoli Access Manager Web Portal Manager

The Tivoli Access Manager Web Portal Manager is a Web-based administration tool used to manage Tivoli Access Manager as an alternative to the **pdadmin** command line interface. This section explains how to install and configure the Web Portal Manager.

Installing the Web Portal Manager

In our example, we installed the Web Portal Manager rpm and then serviced the rpm as part of the base installation. To verify that you have the proper rpm installed, enter the following command:

```
# rpm -qa | grep PDWPM
```

You should see the following result at fixpack 4:

```
PDWPM-PD-5.1.0-4
```

Configuring the Web Portal Manager

To configure the Web Portal Manager, follow these steps:

1. Open a console window.
2. Log on as a root user.
3. Change to the /opt/PolicyDirector/sbin directory.
4. Enter the following command to configure the Web Portal Manager:

   ```
   # ./amwpmcfg -action config -interactive
   ```

   a. Enter the WebSphere Application Server installation path. We entered `/opt/WebSphere/AppServer`. Then press Enter.

   b. Enter the host name of the Tivoli Access Manager Policy Server. We entered `tamlx1`. Then press Enter.

   c. Enter the port number for the Tivoli Access Manager Policy Server. We entered `7135`. Then press Enter.

   d. Enter the Tivoli Access Manager administrator. We entered `sec_master`. Then press Enter.

   e. Enter the password for the Tivoli Access Manager administrator. We entered the password. Then press Enter.

The configuration takes several minutes. The Web Portal Manager application is deployed to the WebSphere Application Server server1.

Verifying the Tivoli Access Manager Web Portal Manager

To verify the Tivoli Access Manager Web Portal Manager, complete these tasks:

1. Restart the following services to acquire the configuration settings for the newly deployed Web Portal Manager:

   a. Start the IBM HTTP Server:

   ```
   # cd /opt/IBMHTTPServer/bin
   # ./apachectl start
   ```

   b. Start the WebSphere Application Server - server1:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./startServer.sh server1
   ```

2. Enter the following URL in a Web browser to access the Web Portal Manager:

   `http://`*`hostname`*`/pdadmin`

   Here *hostname* is the host name where the Web Portal Manager is deployed (for example, `http://tamlx1.itso.ral.ibm.com/pdadmin`).

3. On the Access Manager Login page, complete the following items:

   a. For Secure Domain, accept the default.
   b. For User ID, type `sec_master`.
   c. Specify a password.
   d. Click **Login**.

4. Review the configuration options. To log out, click **Sign off** in the lower right corner of the page.

2.3.13 Verifying the Tivoli Access Manager servers

After you install the fixpack, we recommend that you verify that the Tivoli Access Manager and supporting servers are working properly.

Stopping and starting the Tivoli Access Manager

To stop and start the Tivoli Access Manager Policy and Authorization servers, use the following actions:

- Check the status of the Access Manager Policy and Authorization Server:

  ```
  # /opt/PolicyDirectory/bin/pd_start status
  ```

- Stop the Access Manager Policy and Authorization Server:

  ```
  # /opt/PolicyDirectory/bin/pd_start stop
  ```

- Start the Access Manager Policy and Authorization Server:

  ```
  # /opt/PolicyDirectory/bin/pd_start start
  ```

Starting servers for administration on Policy Server node

To start server for administration purposes on the Policy Server node, complete the following steps:

- Start the IBM HTTP Server for administration purposes:

```
# /opt/IBMHttpServer/bin/apachectl start
```

- Start the WebSphere Application Server server1 for administration purposes:

```
# /opt/WebSphere/AppServer/bin/startServer.sh server1
```

Verifying the Web-based administration tools

To verify the Web-based administration tools, follow these steps:

- For Tivoli Access Manager Web Portal Manager, go to:

```
http://tamlx1.itso.ral.ibm.com/pdadmin
```

- For WebSphere Application Server Administration Console, go to:

```
http://tamlx1.itso.ral.ibm.com:9090/admin
```

- For Tivoli Directory Server Web Administration Tool, go to:

```
http://tamlx1.itso.ral.ibm.com:9080/IDSWebApp/IDSjsp/Login.jsp
```

The installation and base configuration for the Policy Server node components are complete.

2.4 Installing the Reverse Proxy node

This section describes the procedure that we used to install and configure the Reverse Proxy node for the ITSO working example runtime environment on Linux. You can install the Reverse Proxy the Installation Wizards or Native Install Utilities. As mentioned earlier, we decided to use the Installation Wizards to install the product.

> **Note:** When installing and configuring the Reverse Proxy node, we referenced the following product guides and IBM Redbook:
>
> - *Web Security Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1361
> - *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359
> - *A Secure Portal Using WebSphere Portal V5 and Tivoli Access Manager V4.1*, SG24-6077

The high-level tasks to install the Reverse Proxy node require:

1. Installing SUSE LINUX
2. Reverse Proxy node prerequisites
3. Installing IBM Java Runtime Environment
4. Installing IBM GSKit.
5. Installing Tivoli Directory Client
6. Installing Tivoli Access Manager WebSEAL
7. Configuring Tivoli Access Manager WebSEAL
8. Installing Tivoli Access Manager V5.1 base Fixpack 4
9. Installing Tivoli Access Manager V5.1 WebSEAL Fixpack 4

2.4.1 Installing SUSE LINUX

For details about installing SUSE LINUX, refer to “Installing SUSE LINUX” on page 200.

2.4.2 Reverse Proxy node prerequisites

Prior to installing the Policy Server node, ensure that you have completed the following prerequisite steps.

1. Verify that you have enough disk space to install the software components on the Directory Server node (refer to Table 2-9). Enter the following command to check the available disk space:

   ```
   df -H
   ```

Table 2-9 Reverse Proxy node disk space

Software	Disk space	Target directory
IBM GSKit V7.0.1.16	10 MB	/usr/local/ibm/gsk
IBM Tivoli Directory Server V5.2 – Client SDK	70 MB	/usr/ldap
IBM Tivoli Access Manager V5.1.0.2 – Runtime – Web Security – Java Runtime Environment (PDJRTE) – WebSEAL	40 MB	/opt/PolicyDirector
Tivoli Directory Server V5.2 – Client	70 MB	/usr/ldap

2. Verify that you have the following Perl packages installed on the Directory Server node. These packages are needed for YaST2 to work properly after

installing service pack 3 (assuming SUSE LINUX V8 Service Pack 3 is installed):

```
perl-XML-DOM-1.39-95.i386
perl-XML-RegExp-0.03-380.i386
```

Check for these packages by entering the following command:

```
# rpm -qa | grep perl-XML
```

For details, see “Installing and updating Perl” on page 206.

2.4.3 Installing IBM Java Runtime Environment

The JRE is needed by the *iKeyman* utility installed with the GSKit. iKeyman is used to create the keystore and create certificates. For details, see “Configuring IBM Java Runtime Environment V1.3.1” on page 207.

2.4.4 Installing IBM GSKit

In our example, we installed the IBM GSKit V7.0.1.16 on the Reverse Proxy node. For details, refer to 2.2.5, “Installing IBM GSKit” on page 25.

2.4.5 Installing Tivoli Directory Client

As a prerequisite to the Tivoli Access Manager Runtime, you must install the Tivoli Directory Server - Client SDK.

1. Move the existing openldap-client files. For details, refer to “Moving the openldap-client files” on page 29.
2. Install the Tivoli Directory Client.
   a. Install the Tivoli Directory Client V5.2. However select only the **Client SDK** option. For details, see 2.2.6, “Installing Tivoli Directory Server V5.2” on page 28.
   b. Install the Tivoli Directory Server V5.2 Fixpack 1. For details, see 2.2.7, “Installing Tivoli Directory Server V5.2 Fixpack 1” on page 33.

2.4.6 Installing Tivoli Access Manager WebSEAL

This section explains how to install the Tivoli Access Manager WebSEAL on the Reverse Proxy node. We use the native installer.

To install the Tivoli Access Manager WebSEAL and supporting packages on the Reverse Proxy node using the native installer, follow this sequence:

1. Ensure that the following prerequisites are met:

   a. Ensure that the Tivoli Directory Server is started on the Directory Server node. For details, see “Stopping and starting the Tivoli Directory Server” on page 36.

   b. Ensure the Tivoli Access Manager Policy Server and Authorization Servers are started on the Policy Server node. For details, see “Stopping and starting the Tivoli Access Manager” on page 56.

   c. Ensure that the JRE is installed, which is required by the iKeyman utility used by GSKit to create a keystore and certificates.

   d. Ensure that the IBM GSKit V7.0.1.16 is installed.

2. Insert the *IBM Tivoli Access Manager v5.1 Web Security* CD.

3. Open a console window and log on as a root user.

4. Mount the CD-ROM:

   ```
   # mount /dev/cdrom /media/cdrom
   ```

5. Change to the directory of the Tivoli Access Manager rpms found on the CD:

   ```
   # cd /media/cdrom/xSeries
   ```

6. Enter the following command to install the Tivoli Access Manager rpms:

   ```
   # rpm -ivh PDRTE-PD-5.1.0-0.i386.rpm PDWebRTE-PD-5.1.0-0.i386.rpm
   PDWeb-PD-5.1.0-0.i386.rpm PDJrte-PD-5.1.0-0.i386.rpm
   ```

   The rpms we installed are defined in Table 2-8.

Table 2-10 Tivoli Access Manager V5.1 rpms for Reverse Proxy node

Tivoli Access Manager rpm	Description
PDRTE-PD-5.1.0-0.i386.rpm	Access Manager Runtime
PDWebRTE-PD-5.1.0-0.i386.rpm	Access Manager Web Security
PDWeb-PD-5.1.0-0.i386.rpm	Access Manager WebSEAL
PDJrte-PD-5.1.0-0.i386.rpm	Access Manager Java Runtime Environment

7. Unmount the CD-ROM:

   ```
   # cd /
   # umount /media/cdrom
   ```

2.4.7 Installing Tivoli Access Manager V5.1 base Fixpack 4

The Reverse Proxy node includes the Tivoli Access Manager V5.1 Runtime and JRE base packages. It must be serviced with Tivoli Access Manager V5.1 Base Fixpack 4.

To install Tivoli Access Manager V5.1 Base Fixpack 4, complete these steps:

1. Ensure that the Tivoli Access Manager WebSEAL is stopped before you install the fixpack:

   ```
   # cd /opt/PolicyDirectory/bin
   # ./pd_start stop
   ```

2. Download Tivoli Access Manager V5.1 Base Fixpack 4 (5.1-TAM-FP04_LIN.tar) from the following Web site to a temporary directory (for example, /tmp/tam51-base.fp4) on the Policy Server node:

   http://www.ibm.com/support/docview.wss?rs=638&context=SSPREK&uid=swg24007368&loc=en_US&cs=utf-8&lang=en

3. Open a console window as root.
4. List the Tivoli Access Manager packages that are currently installed:

   ```
   # rpm -qa | grep PD
   ```

   Verify that the following packages are listed:

   ```
   PDRTE-PD-5.1.0-0
   PDJrte-PD-5.1.0-0
   ```

5. Change to the temporary directory where you downloaded the fixpack (for example, `/tmp/tam51-base.fp4`).
6. Unpack the fixpack as follows from the temporary directory (for example, `/tmp/tam51-base.fp4`):

   ```
   # tar -xvf 5.1-TAM-FP04_LIN.tar
   ```

7. Run the following command on one line to update each package to the 5.1.0.4 level:

   ```
   # rpm -Uvh --noscripts PDRTE-PD-5.1.0-4.i386.rpm PDJrte-PD-5.1.0-4.i386.rpm
   ```

   > **Note:** If Tivoli Access Manager is already configured, you may need to install the package with the **`--noscripts`** flag:
   >
   > ```
   > rpm -U --noscripts <patchname>
   > ```

8. Verify that the packages were updated correctly:

   ```
   # rpm -qa | grep PD
   ```

Verify that the packages were updated to 5.1.0-4:

```
PDJrte-PD-5.1.0-4
PDRTE-PD-5.1.0-4
```

2.4.8 Installing Tivoli Access Manager V5.1 WebSEAL Fixpack 4

This section explains how to install the Tivoli Access Manager V5.1 WebSEAL Fixpack 4. For more information about the contents of Fixpack 4, refer to the readme file located on the Web at:

http://www.ibm.com/support/docview.wss?rs=638&context=SSPREK&uid=swg24007369&loc=en_US&cs=utf-8&lang=en

To install Tivoli Access Manager V5.1 WebSEAL Fixpack 4, follow these steps:

1. Ensure that all prerequisites are met before you install Tivoli Access Manager V5.1 WebSEAL Fixpack 4:

   ```
   # pdweb stop
   ```

2. Download the Tivoli Access Manager V5.1 WebSEAL Fixpack 4 from the Tivoli support site at:

   http://www.ibm.com/support/docview.wss?rs=638&context=SSPREK&uid=swg24007369&loc=en_US&cs=utf-8&lang=en

3. Extract the fixes to the temporary directory (for example, /tmp/tam51-webseal.fp4) using the following command:

   ```
   # tar -xvf 5.1-AWS-FP04-LIN.tar
   ```

4. From the temporary directory, issue the following command to install the WebSEAL Fixpack 4:

   ```
   # rpm -Uvh PDWebRTE-PD-5.1.0-4.i386.rpm PDWeb-PD-5.1.0-4.i386.rpm
   ```

   Note: If Tivoli Access Manager is already configured, you may need to install it with the **`--noscripts`** flag:

   ```
   rpm -U --noscripts <patchname>
   ```

5. After the installation is completed, verify that the packages have updated successfully:

   ```
   # pdversion
   ```

   Check to see that both the Tivoli Access Manager Runtime and the Tivoli Access Manager WebSEAL Server show V5.1.0.4.

2.4.9 Configuring Tivoli Access Manager WebSEAL

This section explains how to configure the Tivoli Access Manager packages that support WebSEAL on the Reverse Proxy node. The following tasks are required:

1. Configuring the Tivoli Access Manager Runtime
2. Configuring the Tivoli Access Manager WebSEAL
3. Configuring Tivoli Access Manager Java Runtime Environment
4. Checkpoint: Verifying the WebSEAL installation

Configuring the Tivoli Access Manager Runtime

To configure the Tivoli Access Manager Runtime using the **`pdconfig`** utility, complete the following steps:

1. Open a console window as a root user.
2. Start the **`pdconfig`** utility:

   ```
   # pdconfig
   ```

3. From the Tivoli Access Manager Setup Menu, enter `1` (Configure Package) and press Enter.
4. In the Tivoli Access Manager Configuration Menu, enter `1` (Access Manager Runtime Configuration) and press Enter.
5. On the next window, respond to the following prompts as indicated:

   a. Will the Policy Server be installed on this node y/n? `No`
   b. Do you want to configure Tivoli Common Directory logging y/n? `No`
   c. Registry type: `LDAP`
   d. LDAP server host name: `ldaplx1.itso.ral.ibm.com`
   e. LDAP server port: `389`
   f. Policy Server host name: `tamlx1.itso.ral.ibm.com`
   g. Policy Server SSL port: `7135`
   h. Domain: `Default`
   i. Automatically download the pdcacert.b64 file from the policy server?: `Yes`

Configuring the Tivoli Access Manager WebSEAL

To configure the Tivoli Access Manager WebSEAL using the **pdconfig** utility, use a similar process as in "Configuring the Tivoli Access Manager Runtime" on page 63.

We provide the following ITSO working example sample values used to configure the Tivoli Access Manager WebSEAL:

- WebSEAL instance name: `Default`
- Use logical network interface: `No`
- WebSEAL host name: `wslx1`

> **Note:** When the instance is created, the server name is generated as *`instance_name`*`-webseald-`*`hostname`*, for example, `default-webseald-wslx1`.

- WebSEAL listening port: `7234`
- Administrator ID: `sec_master`
- Administrator password: `<password>`
- Enable SSL communication with LDAP server y/n?: `No`
- Allow HTTP access y/n?: `Yes`
- HTTP Port: `80`
- Allow secure HTTPS access y/n?: `Yes`
- HTTPS Port: `443`
- Web document root: `/opt/pdweb/www-default/docs` (default)

Configuring Tivoli Access Manager Java Runtime Environment

To configure the Tivoli Access Manager Java Runtime Environment (PDJRTE) using the **pdconfig** utility, use a similar process as in "Configuring the Tivoli Access Manager Runtime" on page 63.

We provide the following ITSO working example sample values used to configure the Tivoli Access Manager Java Runtime Environment (PDJRTE):

- Full path of the JRE: `/usr/lib/IBMJava2-1.3.1/jre`
- Full or stand-alone: `Stand-alone`
- Tivoli Common Directory Logging: `No`

Stopping and starting WebSEAL

To stop and start WebSEAL, perform the following actions:

1. Check the status of WebSEAL:

   ```
   /opt/PolicyDirector/bin/pd_start status
   ```

2. Stop WebSEAL:

   ```
   /opt/PolicyDirector/bin/pd_start stop
   ```

3. Start WebSEAL:

```
/opt/PolicyDirector/bin/pd_start start
```

Checkpoint: Verifying the WebSEAL installation

After you install the Tivoli Access Manager V5.1 WebSEAL and Web Security runtime, verify that they work as expected.

1. Ensure that WebSEAL is started.
2. Enter the following URL in a Web browser to access WebSEAL on the Reverse Proxy node:

   ```
   http://wslx1.itso.ral.ibm.com
   ```

3. You should see a window from WebSEAL with `Forbidden` written in large red letters. This window identifies unexpected errors such as you would see when unauthorized users try to access a site.

   In the center of the page, click the **Re-access the page using HTTPS** link to re-access the page.

4. While trying to access the page using Secure Sockets Layer (SSL), the page produces an error message. This page appears because the certificate that ships with WebSEAL is a self-signed certificate. Therefore, it is not signed by a recognized Certificate Authority. To continue the test, select **Accept this certificate permanently** and then click **OK**.
5. Not only is the test certificate that ships with WebSEAL self-signed, it was also issued with another server name. Now you see a window with this information. Click **OK**.
6. When prompted to login, provide the user name and password as set during the installation of WebSEAL.
   a. For User name, specify `sec_master`.
   b. Enter a password.

7. You now see the default WebSEAL home page (Figure 2-6). Close all browser windows that are open. This is the only way to log out of a system that is using Basic Authentication. The credentials are cached at the browser and are deleted only when *all* browser windows are closed.

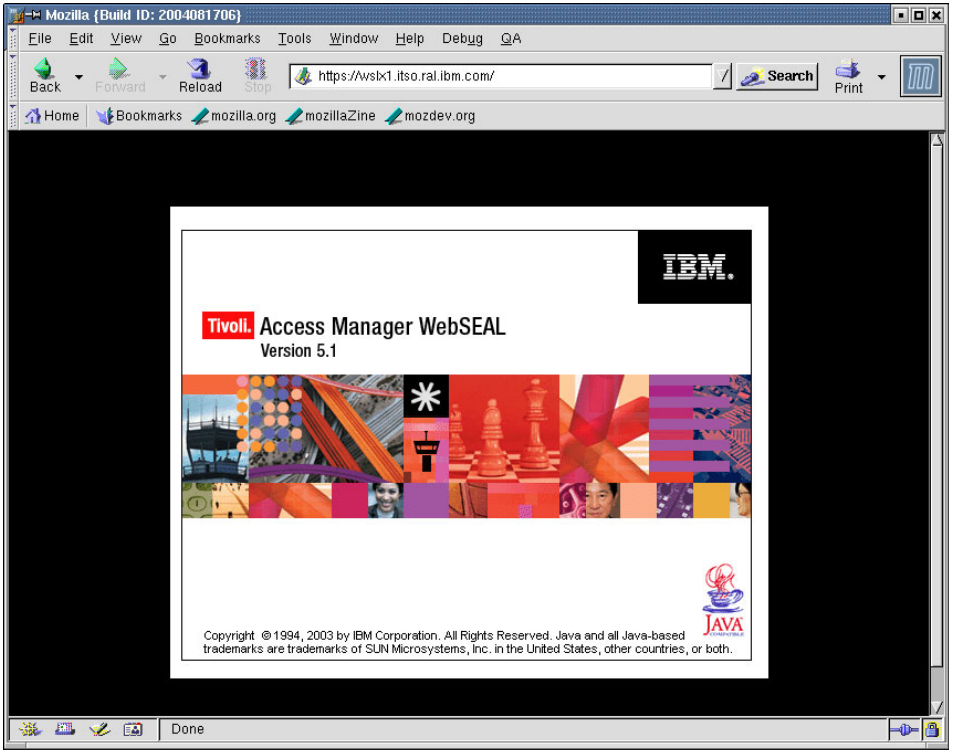

Mozilla {Build ID: 2004081706}
File Edit View Go Bookmarks Tools Window Help Debug QA
Back Forward Reload Stop
https://wslx1.itso.ral.ibm.com/
Search
Print
Home Bookmarks mozilla.org mozillaZine mozdev.org
IBM.
Tivoli. Access Manager WebSEAL
Version 5.1
Copyright ©1994, 2003 by IBM Corporation. All Rights Reserved. Java and all Java-based trademarks are trademarks of SUN Microsystems, Inc. in the United States, other countries, or both.
JAVA
Done

Figure 2-6 Tivoli Access Manager WebSEAL home page

This completes the base installation and base configuration of the Reverse Proxy node.

2.5 Installing the Portal Server node

This section describes the procedure that we used to install and configure the Portal Server node for the ITSO working example runtime environment on SUSE LINUX.

Note: When installing and configuring the Portal Server node, we referenced the following resources:

- *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325
- IBM WebSphere Portal Extend for Multiplatforms V5.0.2 Information Center:

  `http://www.ibm.com/websphere/portal/library`

The following high-level tasks are required to install the Portal Server node:

1. Installing SUSE LINUX
2. Portal Server node prerequisites
3. Installing WebSphere Portal Server V5.0.2.2
4. Installing IBM GSKit
5. Installing Tivoli Directory Client
6. Installing Tivoli Access Manager Runtime and PDJRTE
7. Installing Tivoli Access Manager V5.1 Base Fixpack 4
8. Configuring Tivoli Access Manager Runtime and PDJRTE
9. Installing DB2 Universal Database client
10. Configuring the DB2 UDB client/server node

2.5.1 Installing SUSE LINUX

For details about installing SUSE LINUX Enterprise Server V8, refer to “Installing SUSE LINUX” on page 200.

2.5.2 Portal Server node prerequisites

Verify that you have enough disk space to install the software components on the Portal Server node (refer to Table 2-11). Enter the following command to check the disk space that is available:

```
df -H
```

Table 2-11 Portal Server node disk space

Software	Disk space	Target directory
IBM WebSphere Portal V5.0.2.2	1.5 GB	/opt/WebSphere/PortalServer
IBM WebSphere Application Server Enterprise V5.0.2.6	900 MB	/opt/WebSphere/AppServer
IBM HTTP Server V1.3.26	30 MB	/opt/IBMHTTPServer

Software	Disk space	Target directory
IBM DB2 UDB V8.1 Client	400 MB	/opt/IBM/db2/V8.1
IBM DB2 UDB V8.1 Fixpack 6a	700 MB	/tmp (the location of tar file)
Tivoli Directory Server V5.2 – Client	70 MB	/usr/ldap
IBM Tivoli Access Manager V5.0.1.4 – Runtime – Java Runtime Environment (PDJRTE)	40 MB	/opt/PolicyDirector

2.5.3 Installing WebSphere Portal Server V5.0.2.2

This section explains how to install and configure WebSphere Portal Server V5.0.2.2 on the Portal Server node. The WebSphere Portal Installer installs the following components:

- WebSphere Application Server Enterprise (Base and PME)
- IBM HTTP Server
- WebSphere Portal Server
- WebSphere Portal Content Publisher

The following high level steps are required to install WebSphere Portal V5.0.2.2:

1. WebSphere Portal installation prerequisites
2. Installing WebSphere Portal V5.0.2.2
3. Verifying the WebSphere Portal installation

WebSphere Portal installation prerequisites

You must meet the following prerequisites for the WebSphere Portal installation:

- Table 2-12 lists the required CDs needed to install IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 for the ITSO working example. If you want to install WebSphere Portal for Linux from a network or file system from unpacked CDs, note the directory names in Table 2-12. If you use the directory names provided, the WebSphere Portal Installer continues without prompting for the CDs.

Table 2-12 IBM WebSphere Portal Extend for Multiplatforms V5.0.2 CDs used for ITSO runtime

CD image directory name	CD description
setup	WebSphere Portal Installer
cd1-3	WebSphere Application Server Enterprise V5.0 for Linux
cd1-8	WebSphere Application Server PTF and Cumulative Fixes for Linux – WebSphere Application Server 5 Base Fixpack 2 (V5.0.2) – WebSphere Application Server 5 PME Cumulative Fix 2 (V5.0.2.2) – WebSphere Application Server 5 Base Cumulative Fix 6 (V5.0.2.6)
cd2	WebSphere Portal Server V5.0.2.2 – Personalization

- Disk space is needed for WebSphere Portal and supporting components. See Table 2-6 on page 17 for details.

Installing WebSphere Portal V5.0.2.2

To install WebSphere Portal V5.0.2.2, follow these steps:

1. Insert the *IBM WebSphere Portal V5.0.2.2 Setup* CD.
2. Open a terminal window, and log on as a root user.
3. Navigate to the *CD_root*, and enter the following command to start the WebSphere Portal Installer:

   ```
   # ./install.sh
   ```

4. In the Install Shield Language window, select the desired language (for example, English) and click **OK**.
5. In the Welcome window, you are provided with an option to launch the Information Center (optional). Click **Next** to continue the installation.
6. When you see the License Agreement panel, review the terms and, if you are in agreement, select **I accepts the terms in the license agreement**. Then click **Next**.

   The installer checks for the required operating system and prerequisites.

7. The next window enables you to choose a Full (all components) or Custom (useful when components such as WebSphere Application Server are already installed) installation. In our example, we selected **Full**. Click **Next**.
8. In the window to specify the WebSphere Application Server installation directory, specify a directory. We used the path `/opt/WebSphere/AppServer`. Click **Next**.

9. The next window displays the IBM HTTP Server installation directory. We used the path `/opt/IBMHTTPServer`. Click **Next**.

10. The next window prompts for the node name and host name to be used for the WebSphere Application Server install. We used the following values:

    - Node name: `wps11x`
    - WebSphere Application Server host name: `wps11x.itso.ral.ibm.com`

    Click **Next**.

11. You are now prompted for the WebSphere Portal installation directory. We used the path `/opt/WebSphere/PortalServer`. Click **Next**.

12. You are prompted for the WebSphere Portal administrator user ID and password (used to logon after installation). We entered the following values:

    - WebSphere Portal administrative user: `wpsadmin`
    - WebSphere Portal administrative user password: `<password>`
    - Confirm password: `<password>`

    Click **Next**.

13. The next window displays the different components that are going to be installed. Click **Next**.

14. The Installer program prepares the installation. After a while, you are prompted to insert CD #1-1 *WebSphere Application Server Enterprise for Windows*. It begins by locating a Java Virtual Machine, and then installs WebSphere Application Server (Base and PME).

    After this installation is completed, you are prompted to insert CD #1-6 *WebSphere Application Server Fixpack and eFixes for Windows and Linux*.

    > **Note:** For this example, we avoided the necessity of replacing CDs by putting the CD images in the network drive. By verifying that you use the following directory names for each CD image (and place the images in the same directory in the network), you are not prompted to insert any CDs:
    >
    > - setup
    > - cd1-1
    > - cd1-6
    > - cd2

    The wizard performs the following tasks, displaying a progress meter for each task:

    - Preparing the WebSphere Application Server Fix Pack files
    - Installing WebSphere Application Server Fix Pack 1
    - Installing WebSphere Application Server Enterprise Fix Pack 1
    - Installing WebSphere Application Server Fixes

15. When these tasks are complete, the installer starts the WebSphere Application Server (server1 application server). After the server starts, you are prompted to insert CD #2 *WebSphere Portal Server - WebSphere Portal Content Publisher*. It starts installing WebSphere Portal.
16. When the installation is complete, click **Finish**.

Verifying the WebSphere Portal installation

To verify that WebSphere Portal and WebSphere Application Server were installed properly, perform the steps in the following sections.

Verifying the WebSphere Portal Server

To verify the WebSphere Portal is working properly, complete these tasks:

1. Open a console window as a root user on the Portal Server node.
2. Change to the following directory:

   ```
   # cd /opt/WebSphere/AppServer/bin
   ```

3. Start the WebSphere Portal Server:

   ```
   # ./startServer.sh WebSphere_Portal
   ```

4. Enter the following URL in a Web browser to access the WebSphere Portal home page:

   ```
   http://wpslx1.itso.ral.ibm.com:9081/wps/portal
   ```

   > **Note:** By default after installation, the WebSphere Portal Server is not configured to use an external Web server. Therefore, we must specify the port (9081).

5. Click the **Log in** link located in the upper right corner to login to the portal. You see new page that prompts you for login information.
   a. For User ID, type `wpsadmin`. This user ID was created during the WebSphere Portal installation.
   b. Enter your password. The password was user defined during the WebSphere Portal installation.
6. You should see the personalized portal pages for the logged in user. If the installation is successful and you have Internet access, you should see a portal home page.

   If your computer does not have direct access to the Internet, you see a page with broken links. This error is purely cosmetic and happens because some of the given portlets obtain content to display by connecting to servers that are available on the Internet.

7. Click **Log out**.

Verifying the WebSphere Application Server

To verify the WebSphere Application Server, follow these steps:

1. Open a console window as a root user on the Portal Server node.
2. Change to the following directory:

   ```
   # cd /opt/WebSphere/AppServer/bin
   ```

3. Start the server1 application server:

   ```
   # ./startServer.sh server1
   ```

   > **Note:** By default, the WebSphere Administrative Console Enterprise application is installed to the application server named server1.

4. Enter the following URL in a Web browser to access the WebSphere Application Server Administrative Console:

   `http://wpslx1.itso.ral.ibm.com:9090/admin`

5. Log in as `wasadmin`. At this stage of the configuration, WebSphere security is not enabled. Therefore, no password is required.
6. After reviewing the configuration options, click **Logout**.

2.5.4 Installing IBM GSKit

On the Portal Server node, only the Tivoli Access Manager JRE is required to be installed. There is a catch. To configure the Tivoli Access Manager JRE, make sure that you have installed Tivoli Access Manager Runtime (**`pdconfig`**), which requires that IBM GSKit is also installed. For details about installing the IBM GSKit, refer to 2.2.5, “Installing IBM GSKit” on page 25.

2.5.5 Installing Tivoli Directory Client

On the Portal Server node, only the Tivoli Access Manager JRE is required to be installed. There is a catch. To configure the Tivoli Access Manager JRE, make sure that you have installed Tivoli Access Manager Runtime (**`pdconfig`**), which requires that the Tivoli Directory Client is also installed.

As a prerequisite to the Tivoli Access Manager Runtime, you must install the Tivoli Directory Server - Client SDK. Follow these steps:

1. Move existing openldap-client files. For details, refer to “Moving the openldap-client files” on page 29.

2. Install the Tivoli Directory Client.

   a. Install the Tivoli Directory Client V5.2. However select only the **Client SDK** option. For details, see 2.2.6, “Installing Tivoli Directory Server V5.2” on page 28.

   b. Install the Tivoli Directory Server V5.2 Fixpack 1. For details, refer to 2.2.7, “Installing Tivoli Directory Server V5.2 Fixpack 1” on page 33.

2.5.6 Installing Tivoli Access Manager Runtime and PDJRTE

Install and configure the Tivoli Access Manager V5.1 Runtime and Java Runtime Environment (PDJRTE) using the native installation utility on the Portal Server node.

1. Prior to installation, ensure that the following are started:

   a. Start the Tivoli Directory Server on the Directory Server node. For details, refer to “Stopping and starting the Tivoli Directory Server” on page 36.

   b. Start the Tivoli Access Manager on the Policy Server node. For details, refer to “Stopping and starting the Tivoli Access Manager” on page 56.

2. Insert the *Tivoli Access Manager Base for Linux* CD.

3. Open a console window as a root user.

4. Mount the CD-ROM:

```
# mount /dev/cdrom /media/cdrom
```

5. Change to the directory of the Tivoli Access Manager rpms found on the CD:

```
# cd /media/cdrom/xSeries
```

6. Enter the following command to install the Tivoli Access Manager JRE rpm:

```
# rpm -ivh PDRTE-PD-5.1.0-0.i386.rpm PDJrte-PD-5.1.0-0.i386.rpm
```

7. Unmount the CD-ROM:

```
# cd /
# umount /media/cdrom
```

2.5.7 Installing Tivoli Access Manager V5.1 Base Fixpack 4

Install Tivoli Access Manager V5.1 Base Fixpack 4. For complete details, refer to 2.4.7, “Installing Tivoli Access Manager V5.1 base Fixpack 4” on page 61.

1. Ensure all Tivoli Access Manager processes are stopped.

2. Run the following command to install Fixpack 4:

```
# rpm -Uvh PDRTE-PD-5.1.0-4.i386.rpm PDJrte-PD-5.1.0-4.i386.rpm
```

Note: If Tivoli Access Manager is already configured, you may need to install it with the **--noscripts** flag:

```
rpm -U --noscripts <patchname>
```

2.5.8 Configuring Tivoli Access Manager Runtime and PDJRTE

This section explains how to configure the Tivoli Access Manager Runtime and PDJRTE.

Configuring Tivoli Access Manager Runtime

To configure the Tivoli Access Manager Runtime using the **pdconfig** utility, complete these steps:

1. Open a console window as a root user.
2. Start the **pdconfig** utility:

   ```
   # pdconfig
   ```

3. From the Tivoli Access Manager Setup Menu, enter 1 (Configure Package) and press Enter.
4. From the Tivoli Access Manager Configuration Menu appears, enter 1 (Access Manager Runtime Configuration) and press Enter.
5. In the next window, complete the following information:
   a. Will the Policy Server be installed on this node y/n? No
   b. Do you want to configure Tivoli Common Directory logging y/n? No
   c. Registry type: LDAP
   d. LDAP server host name: ldaplx1.itso.ral.ibm.com
   e. LDAP server port: 389
   f. Policy Server host name: tamlx1.itso.ral.ibm.com
   g. Policy Server SSL port: 7135
   h. Domain: Default
   i. Automatically download the pdcacert.b64 file from the policy server?: Yes

Configuring Tivoli Access Manager Java Runtime Environment

To configure the Tivoli Access Manager JRE (PDJRTE) using the **pdconfig** utility, follow these steps:

1. Open a console window as a root user.
2. Start the **pdconfig** utility:

   ```
   # pdconfig
   ```

3. From the Tivoli Access Manager Setup Menu, enter 1 (Configure Package) and press Enter.

4. From the Tivoli Access Manager Configuration Menu, enter the Access Manager Java Runtime Environment Configuration.
5. In the next window, complete the following information:
   a. Full path of the JRE: `/opt/WebSphere/AppServer/java/jre`
   b. Full or standalone: `Full`
   c. Policy Server host name: `tamlx1.itso.ral.ibm.com`
   d. Policy Server SSL port: `7135`
   e. Policy Server Domain: `Default`
   f. Tivoli Common Directory Logging: `No`

2.5.9 Installing DB2 Universal Database client

By default, the WebSphere Portal uses the Cloudscape™ database. As part of the ITSO runtime environment, we configure WebSphere Portal to use DB2 UDB instead of Cloudscape (see 3.2, “Configuring WebSphere Portal for DB2 Universal Database” on page 84). In our example, we chose to install the DB2 UDB client on the Portal Server node and share a common DB2 UDB server, which is installed on the Directory Server node, to host the WebSphere Portal Server databases.

Installing the IBM DB2 Universal Database V8.1 client requires the following steps:

1. DB2 UDB client installation prerequisites
2. Installing the DB2 UDB V8.1 client
3. Verifying that id db2inst1 is a member of the db2grp1 group
4. Changing the JDK_PATH database parameter
5. Checkpoint: Testing the DB2 UDB client
6. Installing DB2 UDB V8.1 Fixpack 6a
7. Checkpoint: Verifying that DB2 UDB is still working

DB2 UDB client installation prerequisites

Prior to installing the DB2 UDB client, ensure that the following prerequisites are fulfilled:

1. Verify that the korn shell is installed by entering the following command in a terminal start window (prerequisite of the DB2 installer):

   ```
   # which ksh
   ```

   This command should return the following location of the korn shell:

   ```
   /usr/bin/ksh
   ```

   If you don’t have the korn shell installed, install it before you proceed.

2. Install the IBM Java Runtime Environment V1.3.1. For details, refer to “Configuring IBM Java Runtime Environment V1.3.1” on page 207.

Installing the DB2 UDB V8.1 client

To install the IBM DB2 V8.1 client, complete the following steps:

1. Open a console window and login as a root user.
2. Insert the *DB2 UDB V8.1 Enterprise Server Edition* CD.
3. Mount the CD-ROM:

   ```
   # mount /dev/cdrom /media/cdrom
   ```

4. Navigate to the *cdrom_root*:

   ```
   # cd /media/cdrom
   ```

5. Enter the following command to start the DB2 installer:

   ```
   # ./db2setup
   ```

6. In the DB2 Installer window, click **Install Products**.
7. In the Select the Product to Install window, select **DB2 Administration Client** and then click **Next**.
8. In the Welcome window for the DB2 Setup Wizard, click **Next**.
9. When you see the License Agreement panel, review the agreement and select **Accept** if you are in agreement. Then click **Next**.
10. In the Select the installation type panel, select **Typical** and then click **Next**.
11. In the Setup DB2 Instance panel, select **Create a DB2 instance** and then click **Next**.
12. In the Select user information for the DB2 instance owner panel, complete the following steps:

    a. Select the **New user** radio button.
    b. For User name, type `db2inst1`.
    c. Leave UID blank.
    d. For Group name, type `db2grp1`.
    e. Leave GID blank.
    f. Enter a password.
    g. Confirm the password.
    h. For Home directory, type `/home/db2inst1`.
    i. Click **Next**.

13. In the Start copying files panel, review the selected options and then click **Finish** to begin copying files.
14. When the installation is complete, click the **Status** tab to review the status of the installation. When you are done, click **Finish**.

Verifying that id db2inst1 is a member of the db2grp1 group

After the DB2 UDB client is installed, verify that the db2inst1 user is a member of the db2grp1 group.

1. To determine if the db2inst1 user is a member of the db2grp1 group, enter the following command from a console window as root:

   ```
   # id db2inst1
   ```

   The command should return the following result:

   ```
   gid=102(db2grp1) groups=102(db2grp1), 101(dasadm1)
   ```

2. Enter the following command to list groups db2inst1 as a member:

   ```
   # groups db2inst1
   ```

   The command should return the following result:

   ```
   db2inst1 : db2grp1 dasadm1
   ```

3. Use the following command to list groups root as a member:

   ```
   # groups root
   ```

   The command should return the following result. You may have a different list, but ensure that db2grp1 is listed.

   ```
   root : root bin pkcs11 db2grp1
   ```

   If db2grp1 is not listed, add root to the group by using the graphical tool for administration or edit the /etc/group file. To edit the /etc/group file, enter:

   ```
   # kate /etc/group
   ```

   The command should open a text editor. Make changes to add root to the group db2grp1:

   ```
   db2grp1:x:102:db2inst1,root
   ```

   Close all existing terminal windows for the desktop to inherit the group changes.

4. Edit /etc/profile.local and add the following line to it:

   ```
   . /home/db2inst1/sqllib/db2profile
   ```

   Note: The correct syntax is to type a dot (.) and then a space before /home.

Changing the JDK_PATH database parameter

During the installation of IBM DB2 UDB V8.1, JDK_PATH is set to /opt/IBMJava2-131/jre/bin/java. You must change this path to point to /usr/lib/IBMJava2-1.3.1 for certain Java-dependent DB2 functions to work properly, including the database catalog:

1. Switch to the db2inst1user by running the **su -** root command:

   ```
   # su - db2inst1
   ```

2. Update the JDK_PATH by issuing the following command:

   ```
   > db2 update dbm cfg using JDK_PATH /usr/lib/IBMJava2-1.3.1
   ```

Checkpoint: Testing the DB2 UDB client

We recommend that you verify that DB2 UDB is working properly prior to installing the fixpack.

1. Switch to the root authority by running the **su -** root command.
2. Issue the **xhost** command to open the xhost server for connections from all clients while logged in as a root user:

   ```
   # xhost +
   ```

3. Switch to the db2inst1 user by running the **su -** root command:

   ```
   # su - db2inst1
   ```

4. Export the display to connect to the X server running on your local system:

   ```
   > export DISPLAY=IPADDRESS:0
   ```

5. Echo the display variable to ensure that it is set properly:

   ```
   > echo $DISPLAY
   ```

6. Test that the path has been changed correctly. Type the following command as db2inst1:

   ```
   > db2cc
   ```

   If everything is correct, the DB2 Control Center should load in another window.

Note: If you see an exception that states that you can’t connect to the X11 window, then it is likely you have not set your display variable correctly or your access controls (xhosts).

Installing DB2 UDB V8.1 Fixpack 6a

We installed IBM DB2 UDB V8.1 Fixpack 6a for Linux. We used Fixpack 6a for several reasons. First, both Tivoli Directory Server V5.2 and WebSphere Portal V5.0.2.2 officially support DB2 UDB V8.1 Fixpack 6a. Also, we wanted to use the same version of DB2 UDB Fixpack on all of the nodes for compatibility reasons.

Note: For more information about the contents of the IBM DB2 UDB V8.1 Fixpack 6a, refer the FixpackReadme.txt file which you can find on the Web at:

ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2linuxIA32v8/fixpak/FP6a_MI00093/

To download and install the IBM DB2 UDB V8.1 Fixpack 6a, follow these steps:

1. Download IBM DB2 UDB V8.1 Fixpack 6a from:

   ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2linuxIA32v8/fixpak/FP6a_MI00093/

   We downloaded the FP6a_MI00093.tar, which is Fixpack 66 for the IBM DB2 Universal Database V8.1, Enterprise Server Edition.

2. Unpack the FP6a_MI00093.tar file to a temporary directory (for example, `/tmp/db2v81.fp6`).
3. Change to the directory in which the installation image is located.
4. Enter the `./installFixPak` command to launch the installation.
5. Update the DB2 instance with the new level of DB2. This is a *mandatory* step.
   a. Log on as a root user.
   b. Enter the following command for each DB2 instance:

      ```
      # INSTHOME/instance/db2iupdt iname
      ```

      Here *iname* is the instance name and *INSTHOME* is the installation directory. In our example, we enter:

      ```
      # /opt/IBM/db2/V8.1/instance/db2iupdt db2inst1
      ```

6. The internal DB2 level is 8.1.0.58 after the installation of Fixpack 6a. In our example, we do not have any existing databases that need special attention (rebind DB2 utilities).

   Change to the db2inst1 user by entering the following command:

   ```
   # su - db2inst1
   ```

   Check the db2 version level to verify the fixpack was installed successfully:

   ```
   > db2level
   ```

   Make sure the two entries DB2 v8.1.0.58 and Fixpack 6a are listed.

Checkpoint: Verifying that DB2 UDB is still working

After you finish installing the DB2 UDB V8.1 fixpack, we recommend that you verify that DB2 UDB is still working properly. The easiest way to verify this is to run **db2cc**. For details, see "Checkpoint: Testing the DB2 UDB client" on page 78.

2.5.10 Configuring the DB2 UDB client/server node

This section explains how to configure the DB2 UDB client installed on the Portal Server node, to communicate with the DB2 UDB server installed on the Directory Server node. The configuration is a prerequisite to configuring WebSphere Portal to use DB2 UDB. In our example, the WebSphere Portal databases are on a remote DB2 server.

This section requires the following tasks:

1. Checking the DB2 TCP/IP service name
2. Cataloging the tcpip node
3. Attaching to the remote DB2 server
4. Cataloging the databases
5. Connecting to the database
6. Listing the databases

Checking the DB2 TCP/IP service name

To find the DB2 TCP/IP service name that is configured on the DB2 server instance in the Directory Server node, follow these steps:

1. Open a console on the Directory Server node.
2. Log on as db2inst1:

   ```
   # su - db2inst1
   ```

3. Enter the following command to display the DB2 configuration:

   ```
   > db2 get dbm cfg
   ```

4. Search for the TCP/IP service name (SVCENAME) in the output of the command in 3. For example, the TCP/IP service name for a default installation is `db2c_DB2`.
5. Search the /etc/services file for the service name *db2c_db2inst1*. Record the port number that is used for this service name (for example, 50001).

Cataloging the tcpip node

On the Portal Server node, to catalog the remote tcpip node, open a DB2 command window. Then enter the following DB2 command:

```
db2 catalog tcpip node node_name remote server_name server port_number
```

port_number is the DB2 instance connection port found on the DB2 server node in the /etc/services file. In our example, the connection port for the DB2 server was 50001.

The service name in our services file looked like the following example:

```
db2c_db2inst1 50001/tcp
```

Alternatively, in place of the port number a service name can be used. If a service name is used, the port and service name must be added to the DB2 client system services file so that it can resolve where to find the system.

In this example, we enter:

```
> db2 catalog tcpip node ldaplx1 remote ldaplx1 server 50001
```

Here *50001* is the port number value obtained for the DB2 UDB server (Directory Server node).

Attaching to the remote DB2 server

From a DB2 command window, type the following command to attach using the db2inst1 user ID:

```
db2 attach to node_name user db2_username using db2_user_password
```

In our example, we enter:

```
> db2 attach to ldaplx1 user db2inst1 using <password>
```

Cataloging the databases

Catalog the DB2 UDB server sample database created on the Directory Server node as a test database.

```
db2 catalog db db_name at node node_name
```

In our example, we enter:

```
# su - db2inst1
> db2 catalog db db_name at node ldaplx1
```

Here *db_name* is the database name that you want to catalog. For example, we created a test1 database on the DB2 UDB server.

Connecting to the database

Connect to the DB2 UDB server sample database created on the Directory Server node as a test database (for example, test1).

```
db2 connect to db_name at node node_name
```

In our example, we enter:

```
# su - db2inst1
> db2 connect to db_name using db2inst1 using <password>
```

Here *db_name* is the database name to which you want to connect.

Listing the databases

Notice that the SAMPLE database is remote, when you list the DB2 database with the following command:

```
> db2 list db directory
```

3

Configuring the runtime environment

As an extension of Chapter 2, “Installing the runtime environment” on page 9, this chapter describes the configuration tasks to integrate and secure Tivoli Access Manager and WebSphere Portal components on the four nodes for the ITSO runtime environment. As a review, the four nodes are:

- Reverse Proxy node
- Portal Server node
- Directory Server node
- Policy Server node

3.1 Downloading the ITSO sample configuration scripts

To simplify the configuration, we provide sample configuration files for a secure portal environment on Linux in an additional material Web download. Refer to Appendix C, “Additional material” on page 221, for details.

To access the additional material, download the 9121code.zip file from the following Web site to a temporary directory (for example, /tmp):

ftp://www.redbooks.ibm.com/redbooks/REDP9121

Unzip the /tmp/9121code.zip as follows:

```
# cd /tmp
# unzip 9121code.zip
```

After you unzip the 9121code.zip file to the /tmp directory, you see the following directory structure:

```
/tmp/9121code
/tmp/9121code/config/ldap
/tmp/9121code/config/tam
/tmp/9121code/config/wps
```

Throughout this chapter, we reference configuration scripts from this additional material.

3.2 Configuring WebSphere Portal for DB2 Universal Database

This section describes how to configure and verify WebSphere Portal V5.0.2.2 to use DB2 Universal Database (UDB) V8.1 in place of the default Cloudscape database.

To configure WebSphere Portal to use DB2 UDB V8.1, complete the following steps:

1. Ensure that you installed DB2 UDB V8.1 Server on the Directory Server node (ldaplx1).

   This task should have been completed as part of the runtime installation. For details, see 2.2.3, “Installing DB2 Universal Database” on page 18.

2. Ensure that you have installed DB2 UDB V8.1 Client on the Portal Server node (wpslx1).

   This task should have been completed as part of the runtime installation. For details, refer to 2.5.9, “Installing DB2 Universal Database client” on page 75.

3. Create the required databases on the DB2 server (Directory Server node).

   Remember that, for the ITSO runtime configuration, the DB2 server was installed on the Directory Server node (ldaplx1). This step is required only if you are running your DB2 server on a remote system.

   WebSphere Portal Server allows you to use either one database or multiple databases to store the Portal Server information. We recommend that you create multiple databases to make it easier to do performance tuning and back up the individual databases. For the ITSO runtime environment, we created four databases: wps50, wmm50, wpcp, and fdbk50.

   On the Directory Server node, open a DB2 command window and enter the following commands:

   ```
   # su - db2inst1
   > db2 create db wps50
   > db2 create db wmm50
   > db2 create db wpcp50
   > db2 create db fdbk50
   ```

4. Catalog these databases from the DB2 UDB client on the Portal Server node. Open a DB2 command window and enter the following commands:

   ```
   # su - db2inst1
   > db2 catalog db wps50 as wps50 at node ldaplx1
   > db2 catalog db wmm50 as wmm50 at node ldaplx1
   > db2 catalog db wpcp50 as wpcp50 at node ldaplx1
   > db2 catalog db fdbk50 as fdbk50 at node ldaplx1
   ```

5. Confirm that the databases are cataloged:

   ```
   > db2 list db directory
   ```

6. Open a console window and navigate to the /opt/WebSphere/PortalServer/config directory.

7. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file:

   ```
   # ./WPSconfig.sh backup-main-cfg-file
   ```

   Note: Run the backup configuration procedure prior to further configuration tasks such as configuring security, externalizing the Web server, or transferring the database (as is the case for our example).

   After you run the **`WPSconfig.sh`** command, you should see a backup of the wpconfig.properties file with a time stamp in the /opt/WebSphere/PortalServer/config directory.

> **Note:** The wpconfig.properties file is a configuration input file used by **`wpsconfig`** to load configuration settings for WebSphere Portal (stored in XML files).

This alternative method for backing up and configuring the wpconfig.properties file is the one that is used predominantly by IBM technical professionals in the field:

a. Back up the original wpconfig.properties (for example, wpconfig.prop.orig).

b. Update the configuration of WebSphere Portal Server. Copy the wpsconfig.properties file to a file name that is representative of the type or changes to be made, for example, wpconfig.properties.ldap for LDAP configuration or wpsconfig.properties.db2 for DB2 configuration.

c. Make all the changes to this file and then copy that file back to wpconfig.properties.

d. Run the ./wpsconfig.sh script against wpconfig.properties.

This method provides a good system to keep track of the various changes from step to step without trying to remember timestamps, etc.

8. Export the WebSphere Portal configuration data found in the Cloudscape database by entering the following command:

   ```
   # ./WPSconfig.sh database-transfer-export-linux
   ```

   Then you should see the `BUILD SUCCESSFUL` message.

9. Modify the wpconfig.properties file to configure WebSphere Portal to use DB2 UDB. Refer to Table 3-1 for the configuration settings used in the ITSO example. For a detailed description of each of the keywords, refer to the WebSphere Portal Information Center found on the Web at:

   `http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html`

   When you reach this site, search on `Configuring WebSphere Portal for DB2`.

Table 3-1 WebSphere Portal configuration settings in wpconfig.properties for DB2

Section of wpconfig.properties file	**Keyword**	**ITSO value**
Database properties	DbSafeMode	false
	DbType	db2
	WpsDbName	wps50
	DbDriver	COM.ibm.db2.jdbc.app.DB2Driver
	DbDriverDs	COM.ibm.db2.jdbc.DB2XADataSource
	DbUrl	jdbc:db2:wps50
	DbUser	db2inst1
	DbPassword	<your_dbuser_password>
	DbLibrary	/opt/IBM/db2/V8.1/java/db2java.zip **Note**: Directory is java, not java12
	WpsDsName	wps50DS
	WpsXDbName	wps5TCP
	WpsDbNode	wpsNode
WebSphere Portal content publishing Database properties	WpcpDbNode	wcmNode
	WpcpXDbName	wpcp5TCP
	FeedbackXDbName	fdbk5TCP
	WpcpDbName	wpcp50 **Note**: Separate database
	WpcpDbUser	db2inst1
	WpcpDbPassword	<your_dbuser_password>
	WpcpDbUrl	jdbc:db2:wpcp50
	WpcpDbEjbPassword	<ejb_password> (user defined, default is `ejb`).
	FeedbackDbName	fdbk50
	FeedbackDbUser	db2inst1
	FeedbackDbPassword	<your_dbuser_password>
	FeedbackDbUrl	jdbc:db2:fdbk50

Section of wpconfig.properties file	Keyword	ITSO value
Member Manager properties	WmmDsName	wmmDS
	WmmDbName	wmm50 **Note**: Separate database
	WmmDbUser	db2inst1
	WmmDbPassword	<your_dbuser_password>
	WmmDbUrl	jdbc:db2:wmm50

> **Note:** As an alternative to storing the passwords in the wpconfig.properties file, you can leave the values blank in the file and specify them with a -D parameter when running the WPSconfig.sh script.

10. Save the updated wpconfig.properties file.
11. Validate the configuration properties.

    From the *wp_home*/config directory, enter the following commands to validate the configuration properties:

    ```
    ./WPSconfig.sh validate-database-driver
    ./WPSconfig.sh validate-database-connection-wps -DDbPassword=<password>
    ./WPSconfig.sh validate-database-connection-wmm -DWmmDbPassword=<password>
    ./WPSconfig.sh validate-database-connection-wpcp
    -DWpcpDbPassword=<password>
    ```

    > **Note:** The -D parameter on the validate-database-connection commands is necessary only if you do not have the database passwords in the wpconfig.properties. In the ITSO example, we left off this parameter.

12. Import the database settings.

    If the validation runs correctly, enter the following commands to run the configuration task to import the database settings:

    ```
    ./WPSconfig.sh database-transfer-import
    ```

    When complete, you should see the `BUILD SUCCESSFUL` message.

13. After you import the database tables, perform a reorg check to improve performance on the DB2 server (Directory Server node).

    a. Open a console command window and log on as db2inst1:

    ```
    # su - db2inst1
    ```

b. Run the following commands for each WebSphere Portal database (for example, wps50, wmm50, wpcp50, and fdbk50):

```
db2 connect to database_name
db2 reorgchk update statistics on table all
db2 terminate
db2rbind database_name -l db2rbind.out -u db2inst1 -p <password>
```

c. Repeat this process for other databases.

14. Start the WebSphere Portal server as follows on the Portal Server node:

```
# su - root
# cd /opt/WebSphere/AppServer/bin
# ./startServer.sh WebSphere_Portal
```

15. Verify the WebSphere Portal for DB2 configuration by logging into the portal. From a Web browser window, go to the portal URL:

```
http://wpslx1.itso.ral.ibm.com:9081/wps/portal
```

16. Log on as `wpsadmin` and verify that the portal is loading.

3.3 Configuring WebSphere Portal for IBM HTTP Server

Now we reconfigure WebSphere Portal Server to use an external Web server instead of the internal HTTP service of the WebSphere Application Server. For our example, we used IBM HTTP Server as our external Web server.

> **Note:** This section does not apply to runtime topologies such as the development WebSphere Test Environment or a runtime environment architected not to include an external Web server.

To configure an external IBM HTTP Server for WebSphere Portal in place of the WebSphere Application Server internal HTTP service, follow these steps:

1. Verify that the IBM HTTP Server is started:

   ```
   # ps -ef | grep httpd
   ```

   If it is not started, enter the following command to start the IBM HTTP Server:

   ```
   # /opt/IBMHttpServer/bin/apachectl start
   ```

2. Change to the `/opt/WebSphere/PortalServer/config` directory.
3. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file by entering the following command:

   ```
   # ./WPSconfig.sh backup-main-cfg-file
   ```

4. Change the wpconfig.properties values in Table 3-2. For a detailed description of each of the keywords, refer to the WebSphere Portal Information Center at:

   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html

   When you reach this site, search on `Configuring your Web server`.

Table 3-2 ITSO example wpconfig.properties values for external Web server

Section in wpconfig.properties file	Keyword	ITSO example value
WebSphere Application Server	WpsHostName	wpslx1.itso.ral.ibm.com
	WpsHostPort	80

5. Save the updated wpconfig.properties file.
6. Enter the following command to configure WebSphere Portal Server for the external Web server:

   ```
   # ./WPSconfig.sh httpserver-config
   ```

   When complete, you should see the `BUILD SUCCESSFUL` message.
7. Re-generate the Webserver plug-in.
   a. Start the application server, server1, on the Portal Server node:

      ```
      # /opt/WebSphere/AppServer/bin/startServer.sh server1
      ```

   b. Start the WebSphere Application Server Administrative Console. Enter the following URL in a Web browser:

      ```
      http://wpslx1.itso.ral.ibm.com:9090/admin
      ```

   c. When prompted, log in with any name. Since security is not yet enabled, this username is currently used only for session management.
   d. In the Administrative Console, select **Environment** →**Update Web Server Plugin**.
   e. Click **OK**.

      You should see the `The Web server plugin was updated successfully` message.
   f. Click **Logout**.
8. Restart the IBM HTTP Server by entering the following command in a console window as a root user:

   ```
   # cd /opt/IBMHttpServer/bin
   # ./apachectl -restart
   ```

9. Restart the WebSphere Portal Server:

   a. Open a console window as a root user. Change to the WebSphere Application Server bin directory:

   ```
   # cd /opt/WebSphere/AppServer/bin
   ```

   b. Confirm that the WebSphere Portal Server is stopped:

   ```
   # ./stopServer.sh WebSphere_Portal
   ```

   c. Wait for confirmation that the server instance is stopped or cannot be reached. Then start the WebSphere Portal server:

   ```
   # ./startServer.sh WebSphere_Portal
   ```

10. Verify that WebSphere Portal works properly with the external Web server. You should be able to browse your WebSphere Portal Server using the external Web server host name:

    ```
    http://wpslx1.itso.ral.ibm.com/wps/portal
    ```

> **Note:** Prior to adding the external Web server, you needed to include the port number 9081 for the internal HTTP transport the WebSphere Portal Server was using in the URL, for example:
>
> `http://wpslx1.itso.ral.ibm.com:9081/wps/portal`
>
> Now that we configured the external Web server, we do not need to specify the port (default port 80) in the URL:
>
> `http://wpslx1.itso.ral.ibm.com/wps/portal`

3.4 Configuring WebSphere Portal for LDAP

This section explains how to configure WebSphere Portal with the Tivoli Directory Server (Lightweight Directory Access Protocol (LDAP)). First we connect WebSphere Portal to the previously installed Tivoli Directory Server on the Directory Server node. Then we create users and groups for the ITSO working example environment.

The following tasks are required for this configuration:

1. Creating suffixes
2. Creating the LDIF file containing users and groups
3. Importing the LDIF file (wp-itso.ldif) to create users and groups
4. Verifying the Tivoli Directory Server
5. Configuring WebSphere Portal for read-only LDAP access
6. Enabling LDAP security for WebSphere Portal
7. Verifying the LDAP configuration

Note: For detailed information, see the WebSphere Portal V5.0.2 Information center on the Web at:

http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html

Search for the following topics within the Information Center:

- Setting up IBM Directory Server
- Configuring WebSphere Portal for IBM Directory Server

3.4.1 Creating suffixes

To create the create the root DN suffix for the ITSO working example for the Tivoli Directory Server, follow these steps:

1. Stop the Tivoli Directory Server.

   The configuration tool does not allow configuration changes while the Tivoli Directory Server is running. For details refer to “Stopping and starting the Tivoli Directory Server” on page 36.

2. Start the Tivoli Directory Server Configuration tool on the Directory Server node:

   ```
   # ldapxcfg
   ```

3. Create a DN suffix used as the root suffix for the ITSO working example.

   a. Under Choose a task, select **Manage Suffixes**.

   b. On the Manage Suffixes page, for Suffix DN, enter `dc=itso,dc=ibm,dc=com`.

   c. Click the **Add** button.

4. You should now see the new suffix listed. Click **OK**.

   Note: If you are using IBM Tivoli Directory Server V5.1, refer to the *Base Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1362, for additional steps to prepare the schema.

3.4.2 Creating the LDIF file containing users and groups

Users and groups can be created for the Tivoli Directory Server from the Web Administration Tool or by importing an LDIF file that contains users and groups. Figure 3-1 shows the main structure of the wp-itso.ldif file.

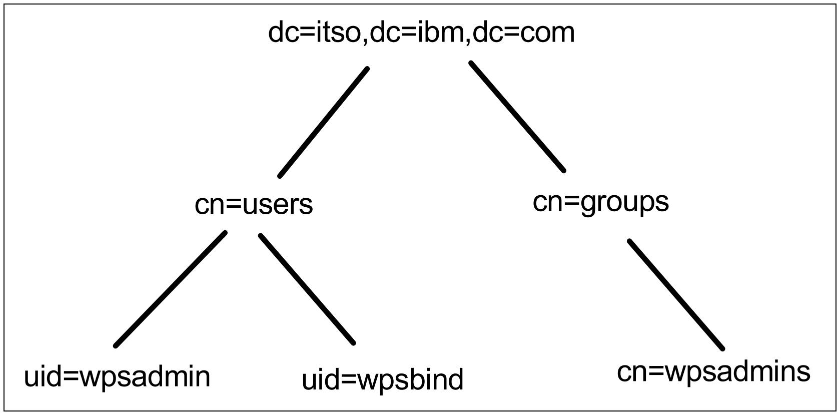


Figure 3-1 LDAP structure to be used in the ITSO example

For the ITSO working example, we created the wp-itso.ldif based on the PortalUsers.ldif file found on the WebSphere Portal Setup CD. The LDIF file is used to import users and groups.

After unpacking the 9121code.zip to the /tmp directory, you will find the wp-itso.ldif file in the directory structure /tmp/9121code/config/ldap/wp-itso.ldif.

Example 3-1 displays the contents of the ITSO sample wp-itso.ldif file. We changed the DN. We recommend that you change the user password attribute for users *wpsadmin* and *wpsbind* to a unique password for your environment.

Note: Ensure the wpsadmin password set in the LDIF file matches the wpsadmin password defined for WebSphere Portal (initially set during the WebSphere Portal installation).

Example 3-1 ITSO example WebSphere Portal LDIF file (wp-itso.ldif)

```
version: 1
# ITSO example: wp-itso.ldif file

dn: dc=itso,dc=ibm,dc=com
objectclass: domain
objectclass: top
# Add lines according to this scheme that correspond to your suffix
dc: itso,dc=ibm,dc=com
```

```
dc: itso,dc=ibm
dc: itso

dn: cn=users,dc=itso,dc=ibm,dc=com
objectclass: container
objectclass: top
cn: users

dn: cn=groups,dc=itso,dc=ibm,dc=com
objectclass: top
objectclass: container
cn: groups

dn: uid=wpsadmin,cn=users,dc=itso,dc=ibm,dc=com
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: wpsadmin
userpassword: wpsadmin
sn: admin
givenName: wps
cn: wps admin

dn: uid=wpsbind,cn=users,dc=itso,dc=ibm,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: wpsbind
userpassword: wpsbind
sn: bind
givenName: wps
cn: wps bind

dn: cn=wpsadmins,cn=groups,dc=itso,dc=ibm,dc=com
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=wpsadmin,cn=users,dc=itso,dc=ibm,dc=com
cn: wpsadmins
```

3.4.3 Importing the LDIF file (wp-itso.ldif) to create users and groups

To import the ITSO-provided wp-itso.ldif file to create users and groups to the Tivoli Directory Server, follow these steps on the Directory Server node:

1. Stop the Tivoli Directory Server. Refer to “Stopping and starting the Tivoli Directory Server” on page 36 for details.
2. Start the Tivoli Directory Server Configuration Tool:

   ```
   # ldapxcfg
   ```

3. Select **Import LDIF data**.
4. On the Import LDIF Data page, complete the following tasks (based on our example):

   a. For Path and LDIF file name, type `/tmp/9121code/config/ldap/wp-itso.ldif`.
   b. Select **Standard import**.
   c. Click **Import** at the bottom of the page.

   After the import finishes successfully, a message is displayed that reports how many entries have been imported into the directory server.

5. When the import is complete, click **Close**.
6. Click **File** →**Close** to close the Tivoli Directory Server Configuration Tool.
7. Restart the Tivoli Directory Server. For details, refer to “Stopping and starting the Tivoli Directory Server” on page 36.

3.4.4 Verifying the Tivoli Directory Server

After you start the Tivoli Directory Server, perform an LDAP search to verify that the LDAP entries (users and groups) were created properly. For example, we entered this command on a console window on the Directory Server node:

```
ldapsearch -h ldaplx1 -b dc=itso,dc=ibm,dc=com -D
uid=wpsbind,cn=users,dc=itso,dc=ibm,dc=com -w wpsbind -s sub uid=wpsadmin
```

Here *-w wpsbind* is the wpsbind user password.

3.4.5 Configuring WebSphere Portal for read-only LDAP access

For the ITSO example, we chose to prevent WebSphere Portal from writing to the LDAP directory. We allow only Tivoli Access Manager to write to the LDAP directory to ensure consistency and increase security. However, the default configuration for WebSphere Portal is not set up for read-only LDAP access (default configuration uses cn=root). Therefore, we must configure WebSphere Portal for read-only LDAP access.

To configure WebSphere Portal for read-only LDAP access, complete these steps:

1. Navigate to the following directory on the Portal Server node:

   ```
   # cd /opt/WebSphere/PortalServer/config/templates/wmm
   ```

2. Back up the following files:

   – wmm_LDAP.xml.*LDAPType.number*.wmm

     For example, `wmm_LDAP.xml.IBM_DIRECTORY_SERVER.1.wmm`

   – wmmLDAPAttributes_*LDAPType*.xml

     For example, `wmmLDAPAttributes_IBM_DIRECTORY_SERVER.xml`

> **Note:** The naming constructs for wmm xml files consist of:
>
> ► *LDAPType*: The type of LDAP server that is being used, such as IBM_DIRECTORY_SERVER or DOMINO502.
>
> ► *number*: Specifies whether a lookaside database is implemented. Use 1 if a lookaside database is not defined or 3 if it is defined.

3. Modify the attributes in the wmm_LDAP.xml.IBM_DIRECTORY_SERVER.1.wmm file.

   Search for the `<ldapRepository` tag section of the file, and modify the values as shown in Example 3-2. We had to add the `ignoreReadOnlyUpdate` attribute.

Example 3-2 ITSO example wmm_LDAP.xml.IBM_DIRECTORY_SERVER.1.wmm

```
<ldapRepository name="wmmLDAP"
<!-- Set wmmGenerateExtId to “false” and add ignoreReadOnlyUpdate with “true” as the value -->
           wmmGenerateExtId="false"
           ignoreReadOnlyUpdate="true"
           supportGetPersonByAccountName="true"
           profileRepositoryForGroups="LDAP1"
           supportTransactions="false"
           adminId="@LDAPAdminUIdXml@"
           adminPassword="@EncryptedLDAPAdminPwd@"
           ldapHost="@LDAPHostName@"
           ldapPort="@LDAPPort@"
           ldapTimeOut="6000"
           ldapAuthentication="SIMPLE"
           ldapType="0"
           groupCacheRefreshInterval="-1">
</ldapRepository>
```

4. Save the file.

5. Modify the attributes in the file wmmLDAPAttributes_IBM_DIRECTORY_SERVER.xml.

   Search for `wmmAttributeName="extId"` and modify the `pluginAttributeName` attribute as shown in Example 3-3.

6. Set the readOnly attribute to `true` in *every* attributeMap tag as shown in Example 3-3. If an `attributeMap` tag does not contain a `readOnly` attribute, you must add it.

Example 3-3 ITSO wmmLDAPAttributes_IBM_DIRECTORY_SERVER.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE repositoryAttributes SYSTEM "wmmAttributesMap.dtd">
<repositoryAttributes repositoryName="wmmLDAP">
<!-- Define which LDAP attribute is mapped to external identifier -->
       <attributeMap   wmmAttributeName="extId"
             <!--ibm-entryUuid is the uniqueID withing IBM Directory Server-->
                        pluginAttributeName="ibm-entryUuid"
                        dataType="String"
                        multiValued="false"
                        readOnly="true"/>

<!-- Define which LDAP attribute is used for storing static group members -->
       <attributeMap   wmmAttributeName="groupMember"
                       pluginAttributeName="@LDAPGroupMember@"
                        applicableMemberTypes="Group"
                        dataType="String"
                        valueLength="1024"
                        multiValued="true"
                        readOnly="true" <!--Add attribute if not defined. -->
                        defaultValue="uid=dummy" />
<!--Continue modifying the rest of the attributeMap tags for readOnly access-->
</repositoryAttributes>
```

7. Save and close the file.

3.4.6 Enabling LDAP security for WebSphere Portal

This section explains how to enable LDAP security for WebSphere Portal. On the Portal Server node, you can customize pre-configured templates to configure WebSphere Portal for LDAP.

1. Open a console window as a root user, and navigate to this directory:

   ```
   # cd /opt/WebSphere/PortalServer/config
   ```

2. Back up the WebSphere Portal configuration properties file:

   ```
   # ./WPSconfig.sh backup-main-cfg-file
   ```

3. Change the wpconfig.properties values in Table 3-3. For a detailed description of the wpconfig.properties for LDAP security configuration with WebSphere Portal, refer to the Information Center at:

   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html

   When you reach this site, search for `Configuring WebSphere Portal for IBM Directory Server`.

Table 3-3 ITSO example wpconfig.properties values for LDAP security

Section in wpconfig. properties file	**Keyword**	**ITSO example value**
WebSphere Application Server Properties	WasUserid	uid=wpsbind,cn=users,dc=itso,dc=ibm,dc=com
	WasPassword	<password>
Portal Configuration Properties	PortalAdminId	uid=wpsadmin,cn=users,dc=itso,dc=ibm,dc=com
	PortalAdminIdShort	wpsadmin
	PortalAdminPwd	<password>
	PortalAdminGroupId	cn=wpsadmins,cn=groups,dc=itso,dc=ibm,dc=com
	PortalAdminGroupIdShort	wpsadmins
WebSphere Portal Security LTPA and SSO Configuration	LTPAPassword	<password>
	LTPATimeout	120
	SSODomainName	itso.ral.ibm.com

Section in wpconfig. properties file	Keyword	ITSO example value
LDAP Properties Configuration	Lookaside	false
	LDAPHostName	ldaplx1.itso.ral.ibm.com
	LDAPPort	389
	LDAPAdminUId	uid=wpsbind,cn=users,dc=itso,dc=ibm,dc=com **Note**: The default is cn=root. In our example, we configured our environment so that WebSphere Portal has read-only access to the LDAP directory. Only Tivoli Access Manager will write to the LDAP directory.
	LDAPAdminPwd	<password>
	LDAPServerType	IBM_DIRECTORY_SERVER
	LDAPBindID	uid=wpsbind,cn=users,dc=itso,dc=ibm,dc=com
	LDAPBindPassword	<password>
Advanced LDAP Configuration	LDAPSuffix	dc=itso,dc=ibm,dc=com
	LdapUserPrefix	uid
	LDAPUserSuffix	cn=users
	LdapGroupPrefix	cn
	LDAPGroupSuffix	cn=groups
	LDAPUserObjectClass	inetOrgPerson
	LDAPGroupObjectClass	groupOfUniqueNames
	LDAPGroupMember	uniqueMember
	LDAPUserFilter	(&(uid=%v)(objectclass=inetOrgPerson))
	LDAPGroupFilter	(&(cn=%v)(objectclass=groupOfUniqueNames))
	LDAPsslEnabled	false

4. Save the updated wpconfig.properties file.

> **Note:** If you are not using an external HTTP server or if the HTTP server resides on the same physical system as WebSphere Portal, set the WpsHostName property in the wpconfig.properties file:
>
> ```
> WpsHostName=wpslx1.itso.ral.ibm.com
> ```

5. Ensure that the Tivoli Directory Server is started.
6. Ensure that server1 and WebSphere_Portal application servers are started:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./serverStatus.sh -all
   ```

   If the application servers are not started, enter the following commands:

   ```
   # ./startServer.sh server1
   # ./stopServer.sh WebSphere_Portal
   ```

7. Change to the following directory as a root user:

   ```
   # cd /opt/WebSphere/PortalServer/config
   ```

8. Enter the following command to validate the modified wpconfig.properties:

   ```
   # ./WPSconfig.bat validate-ldap
   ```

   If an error occurs, review the values in the wpconfig.properties. Typographical errors are quite often the cause of an error on this step. Also review the settings in the LDAP server. Ensure that the LDAP server is actually running.

9. If the validation was successful, enable security by entering the following command:

   ```
   # ./WPSconfig.bat enable-security-ldap
   ```

   If the task completes successfully, you see the `BUILD SUCCESSFUL` message.

Note: You may receive the following error message when running the configuration task to enable security:

```
action-create-deployment-credentials:
[xmlaccess] XMLA0006I: Connecting to URL http://localhost:9081/wps/config
[xmlaccess] XMLA0002I: Reading input file
/opt/WebSphere/PortalServer/config/work/createDeploymentCredentials.xml
[xmlaccess] XMLA0011I: Request was accepted.
[xmlaccess] <?xml version="1.0" encoding="UTF-8" ?>
[xmlaccess] <failure>
[xmlaccess]
com.ibm.wps.command.xml.XmlCommandServlet$AuthorizationException:
XMLC0005E: Authorization for user wpsadmin failed.
[xmlaccess] </failure>
. . . .
BUILD FAILED
file:../config/actions/wps_cfg.xml:289: XMLA0015E: Server response
indicates an error.
```

If you receive this error, it means that one of the following is true:

- LDAPAdminUId does not have write permissions to the LDAP server.
- Your Portal server is not configured for read-only access.

Therefore, you must either give LDAPAdminUId (as defined in the wpconfig.properties) write permissions via the LDAP server administration interface, or you can configure your WebSphere Portal server for read-only access as we have done in the ITSO example in 3.4.5, “Configuring WebSphere Portal for read-only LDAP access” on page 95.

10. Change to the /opt/WebSphere/AppServer/bin directory and enter the following command:

    ```
    # ./stopServer server1 -user wpsbind -password password
    ```

    Here *wpsbind* is the WebSphere Administrator user ID and *password* is the password for the user ID.

    Then enter the following command:

    ```
    # ./startServer server1
    ```

3.4.7 Verifying the LDAP configuration

To verify the WebSphere Portal and LDAP configuration, perform these steps:

1. Verify that WebSphere security is working properly by starting the WebSphere Application Server Administration Console. Log in as user ID `wpsbind`. WebSphere security in this case provides the authentication. If security is not working, you are not able to log in with the wpsbind user ID.

   ```
   http://wpslx1.itso.ral.ibm.com:9090/admin
   ```

2. Verify that WebSphere Portal is working properly with the LDAP configuration and WebSphere security.

   a. If everything works properly, you should be able to browse your WebSphere Portal Server using the fully qualified host name, which is now configured to use LDAP:

   ```
   http://wpslx1.itso.ral.ibm.com/wps/portal
   ```

   > **Important:** Using localhost or just the host name for accessing the portal may cause problems after you configure LDAP security. Always use the fully qualified host name for browsing.

   b. From the WebSphere Portal Welcome page, click **Log in** in the top right corner. For example, we used the wpsadmin user ID and password.

> **Note:** For information about disabling LDAP security for WebSphere Portal, refer to the “Security hardening” chapter in *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

3.5 Enabling mutual SSL between WebSEAL and WebSphere Portal

The ITSO example architecture uses an external IBM HTTP Server for WebSphere Portal on the Portal Server node. This section explains how to configure mutual Secure Sockets Layer (SSL) between WebSEAL on the Reverse Proxy node and the IBM HTTP Server on the Portal Server node.

In our example, we have two fundamental requirements. First, we want to provide access for some pages to unauthenticated users. Second, we do not want to create two separate junctions for authenticated and unauthenticated users.

The section requires the following tasks:

1. Configuring IBM HTTP Server SSL
2. Configuring WebSphere Portal for SSL
3. Exporting the IBM HTTP Server CA certificate
4. Importing IBM HTTP Server certificate into WebSEAL keystore
5. Exporting the WebSEAL certificate
6. Importing WebSEAL certificate into IBM HTTP Server keystore
7. Enabling mutual SSL for the IBM HTTP Server

3.5.1 Configuring IBM HTTP Server SSL

This section explains how to configure SSL for the IBM HTTP Server. It is organized into the following tasks:

- Enabling httpd.conf for SSL
- Copying WebSphere plug-in entries to the httpd.conf
- Creating the IBM HTTP Server keystore
- Verifying IBM HTTP Server

Enabling httpd.conf for SSL

To modify the httpd.conf to enable SSL support, complete the following tasks:

1. Stop the IBM HTTP Server V1.3.26 service.

   a. Check to see if the IBM HTTP Server is running. Open a command window and enter the following command:

   ```
   # ps -ef | grep http
   ```

   b. If the server is running, stop it by using the following command:

   ```
   # cd /opt/IBMHttpServer/bin
   # ./apachectl stop
   ```

2. Back up the original httpd.conf, for example:

   ```
   # cd /opt/IBMHttpServer/conf
   # cp httpd.conf httpd.conf.org
   ```

3. Copy the httpd.conf.sample file (located in the current directory) containing the commented SSL directives to the httpd.conf file (overwriting the existing file), for example:

   ```
   # cp httpd.conf.sample httpd.conf
   ```

4. Open the /opt/IBMHttpServer/conf/httpd.conf file with a text editor.

5. Search for the following lines, uncomment the # symbol, and modify the settings as explained for each line in the following list:

   – Verify the ServerName value, which should include your fully qualified host name (for example, `wpslx1.itso.ral.ibm.com`):

     ```
     ServerName fully_qualified_hostname
     ```

   – Uncomment the IBM SSL module for the given SSL encryption level (56 or 128 bit), for example:

     ```
     LoadModule ibm_ssl_module/libexec/mod_ibm_ssl_128.so
     ```

     > **Note:** 56 or 128 is the appropriate encryption level for your locale, for example, 128 for 128-bit encryption in the U.S. and Canada.

   – Uncomment the addModule line for the IBM SSL module, for example:

     ```
     AddModule mod_ibm_ssl.c
     ```

   – When another Listen statement for a port other than port 80 is added to the HTTPD configuration file, it is necessary to uncomment the following line to listen on port 80 for HTTP:

     ```
     Listen 80
     ```

   – Uncomment the following line to listen on port 443 for HTTPS:

     ```
     Listen 443
     ```

   – Uncomment and update the VirtualHost with your host name as follows:

     ```
     <VirtualHost hostname.domain.com:443>
     ```

     > **Note:** Substitute your fully qualified host name in this line, for example, `<VirtualHost wpslx1.itso.ral.ibm.com:443>`.

   – Uncomment:

     ```
     SSLEnable
     ```

   – Uncomment:

     ```
     </VirtualHost>
     ```

   – Uncomment and update the following line:

     ```
     Keyfile /opt/IBMHttpServer/ssl/keyfile.kdb
     ```

     > **Note:** The keyfile path has been modified to include *ssl* instead of keys, for example:
     >
     > `Keyfile /opt/IBMHttpServer/`**`ssl`**`/keyfile.kdb`

   - Uncomment:

     ```
     SSLV2Timeout 100
     SSLV3Timeout 1000
     ```

6. Save the changes to the httpd.conf file.

Copying WebSphere plug-in entries to the httpd.conf

During the installation of the IBM HTTP Server and WebSphere plug-in, the httpd.conf was updated with the plug-in entries. Now that we copied the httpd.conf.sample file in the previous section to enable the SSL directives, we also need to add the WebSphere plug-in entries to the httpd.conf file.

1. Change to the IBM HTTP Server directory:

   ```
   # cd /opt/IBMHttpServer/conf
   ```

2. Modify the httpd.conf file, and add the following lines for the WebSphere plug-in at the end of the file:

   ```
   LoadModule ibm_app_server_http_module
   /opt/WebSphere/AppServer/bin/mod_ibm_app_server_http.so

   WebSpherePluginConfig /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml
   ```

   > **Tip:** This text is in two separate lines. Within the httpd.conf, file the lines do not wrap.
   >
   > If you back up the httpd.conf.org file, you can cut and paste the plug-in entries to the SSL-enabled httpd.conf.

3. Save and close the httpd.conf file.

Creating the IBM HTTP Server keystore

To create the IBM HTTP Server keystore database used to store certificates, follow these steps:

1. Start the IBM Key Management utility. Open a command window and enter the following command:

   ```
   # gsk7ikm
   ```

2. From the menu bar, click **Key Database File** →**New**.
3. In the New window, complete the following tasks:
   a. For Key Database Type, select **CMS**.
   b. For File name, type `keyfile.kdb`.

c. For Location, type `/opt/IBMHttpServer/ssl`.

> **Note:** This path must match the keyfile path in the httpd.conf file.

d. Click **OK**.

4. In the Password Prompt window, complete the following tasks:

   a. Specify your password (to protect the keystore file).

   b. Select **Set expiration time**. Then enter the number of days before the password expires. If no expiration is required, do not select this option.

   > **Note:** Although an expiration time is not required in a development environment, we recommend that you set an expiration period for all keystores used in production environments.

   c. Select **Stash the password to a file**.

   > **Note:** The IBM HTTP Server accesses the password-protected keystore.

   d. Click **OK**.

5. An information window opens and shows the following message:

```
The password has been encrypted and saved in file:
/opt/IBMHttpServer/ssl/keyfile.sth
```

   Click **OK**.

Creating a certificate for the IBM HTTP Server

For development and testing purposes, we create a new self-signed certificate as explained in the following steps:

1. From the Key Management Utility menu bar, select **Create** →**New Self-Signed Certificate**.

   If you closed the Key Management Utility, you must first open the keystore.

2. Fill in the information on the form and click **OK**. In this example, we entered the following information:

   – Key Label: `WP HTTP Server SSL Key`
   – Common Name: `wpslx1.itso.ral.ibm.com`

   This is the fully qualified host name.

- Organization: `IBM`

> **Note:** The organization name is used when specifying the -D parameter in the **`wp-junction.pd`** command file to create the junction. The value is case sensitive.

3. When you are finished, close the Key Management Utility.

Verifying IBM HTTP Server

After the IBM HTTP Server SSL configuration, we recommend that you complete the following verification tests:

- Starting IBM HTTP Server
- Verifying the IBM HTTP Server

Starting IBM HTTP Server

To start the IBM HTTP Server service, from a command window, change to the *ihs_home*/bin directory:

```
# cd /opt/IBMHttpServer/bin
# ./apachectl start
```

Verifying the IBM HTTP Server

To verify that the IBM HTTP Server is working properly, enter the following URLs in a Web browser after the IBM HTTP Server is started:

- Verify HTTP:

  `http://`*`hostname`*

- Verify HTTPS:

  `https://`*`hostname`*

3.5.2 Configuring WebSphere Portal for SSL

This section explains how to enable SSL for WebSphere Portal on the Portal Server node. In this example, we configure the WebSphere Application Server, where WebSphere Portal is installed, to be SSL enabled. We also update the WebSphere Portal configuration for SSL.

Enabling SSL for the WebSphere Application Server

First you must configure the WebSphere Application Server plug-in for the Web server to forward WebSphere Portal traffic that is received over SSL to WebSphere Application Server (which then forwards the traffic to WebSphere Portal). Then update the virtual host list for WebSphere Application Server to include the correct host name and port number. Finally regenerate the plug-in configuration.

> **Note:** For more information about WebSphere Application Server security, refer to *IBM WebSphere V5.0 Security, WebSphere Handbook Series*, SG24-6573.

To enable SSL for the WebSphere Application Server where WebSphere Portal is installed, complete the following tasks:

1. Ensure the Tivoli Directory Server is started. For details, see “Stopping and starting the Tivoli Directory Server” on page 36.
2. Ensure that the application server, server1, is started on the Portal Server node:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./serverStatus.sh -all -user wpsbind -password <password>
   ```

   If server1 is not started, enter the following command:

   ```
   # ./startServer.sh server1
   ```

3. Start the WebSphere Application Server Administrative Console:
   a. Enter the following URL:

      ```
      https://wpslx1.itso.ral.ibm.com:9043/admin
      ```

   b. Enter the WebSphere administrator credentials (for example, `wpsbind`).
4. Select **Environment** →**Virtual Hosts**.
5. Click **default_host**.
6. Click **Host Aliases**.
7. Add a Host Alias for port `443`, and click **New**.
8. In the New Host Alias page, complete the following tasks:
   a. For Host Name, type *.
   b. For Port, type `443`.
   c. Click **OK**.
9. Click **Save**.
10. On the Save to Master Configuration page, click **Save**.
11. Select **Environment** →**Update Web Server Plugin**.
12. When prompted to update the plug-in configuration file, click **OK**.
13. Log out of the WebSphere Application Server Administrative Console.

Note: In our example, the IBM HTTP Server and plugin-cfg.xml are installed on the same node as the WebSphere Application Server. The plugin-cfg.xml is reloaded with the updated settings based on the RefreshInterval set in the plugin-cfg.xml file. By default, it is set to refresh after 60 seconds. If you want the settings to take effect immediately, restart the IBM HTTP Server.

Enabling SSL in the WebSphere Portal configuration

In our example, since we are using Tivoli Access Manager WebSEAL for authentication, we disable WebSphere Portal login pages in 3.6.11, "Configuring Portal login and logout for use with WebSEAL" on page 137.

To enable SSL in the WebSphere Portal configuration, follow these steps:

1. Navigate to the /opt/WebSphere/AppServer/installedApps/wpslx1/wps.ear /wps.war/WEB-INF directory.
2. Back up web.xml to web.xml.orig.
3. Modify web.xml as shown in Example 3-4. Search on SecurityConstraint_1 to find the appropriate section to modify. Inside the SecurityConstraint_1 definition, change the <transport-guarantee> setting from NONE to CONFIDENTIAL.

   Note: CONFIDENTIAL transport_guarantee only allows HTTPS access to the /myportal/* URLs.

Example 3-4 ITSO modified web.xml for WebSphere Portal

```
<security-constraint id="SecurityConstraint_1">
   <web-resource-collection id="WebResourceCollection_1">
      <web-resource-name></web-resource-name>
      <url-pattern>/myportal/*</url-pattern>
      <http-method>DELETE</http-method>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
      <http-method>PUT</http-method>
   </web-resource-collection>
   <auth-constraint id="AuthConstraint_1">
      <description></description>
      <role-name>All Role</role-name>
   </auth-constraint>
   <user-data-constraint id="UserDataConstraint_4">
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
   </user-data-constraint>
</security-constraint>
```

4. Save and close the web.xml file.
5. Restart the WebSphere_Portal application server.
6. Restart the IBM HTTP Server.

Verifying WebSphere Portal SSL configuration

After the configuration is complete, we recommend that you verify that the WebSphere Portal Server is SSL enabled. Enter the following URL in a Web browser:

```
https://wpslx1.itso.ral.ibm.com/wps/myportal
```

Log in with the `wpsadmin` user ID and password. Verify that the Welcome page is displayed.

3.5.3 Exporting the IBM HTTP Server CA certificate

In our example, we used a self-signed certificate for the IBM HTTP Server. This section explains how to export the Certificate Authority (CA) certificate that is imported into the WebSEAL keystore for the purpose of setting up SSL junctions. Use the following procedure to export the certificate:

1. Start the IBM Key Management utility on the Portal Server node:

   ```
   # gsk7ikm
   ```

2. From the menu bar, select **Key Database File →Open**.
3. Select **/opt/IBMHttpServer/ssl/keyfile.kdb** and click **Open**.

   > **Note:** The key database file, also known as a *keystore* or *keyring*, is defined in the httpd.conf file. Search for Keyfile (for example, Keyfile "/opt/IBMHttpServer/ssl/keyfile.kdb").

4. When prompted, enter the password for the keystore.
5. Under Personal Certificates, select the certificate you created in “Creating a certificate for the IBM HTTP Server” on page 106 (for example, WP HTTP Server SSL Key). Click **Extract Certificate**.
6. In the Extract Certificate to a File window, complete the following tasks:
   a. For Data Type, select **Base64-encoded ASCII data** (default).
   b. For Certificate file name, type `wp_httpd_cert.arm`.
   c. For Location, type `/opt/IBMHttpServer/ssl/`.
   d. Click **OK**.
7. You should see the `The requested action has successfully completed.` message on the status bar. Close the Key Management Utility.

3.5.4 Importing IBM HTTP Server certificate into WebSEAL keystore

In our example, the Java Runtime Environment (JRE), GSKit and WebSEAL are installed on the Reverse Proxy node. By default, the JRE (java.security) does not include the IBM JCE and IBM Certificate Management System (CMS) security providers. As part of our configuration of the GSKit iKeyman utility, we added the IBM JCE and IBM CMS security providers (see “Configuring JCE and CMS extensions needed for IBM GSKit” on page 27).

This section explains how to import the IBM HTTP Server certificate into the WebSEAL keystore on the Reverse Proxy node.

1. Ensure that you have added the IBM CMS security provider to the java.security file. See “Configuring JCE and CMS extensions needed for IBM GSKit” on page 27.
2. Determine the WebSEAL keystore file name and location.
   a. Navigate to the /opt/pdweb/etc directory on the Reverse Proxy node.
   b. Open the webseald-default.conf file in a text editor.
   c. Search webseal-cert-keyfile and record the value (for example, `/var/pdweb/www-default/certs/pdsrv.kdb`).
3. Copy the IBM HTTP Server certificate (for example, `wp_httpd_cert.arm`) to the Reverse Proxy node.
   – From Portal Server node (wpslx1) /opt/IBMHttpServer/ssl directory
   – To Reverse Proxy node (wslx1) /var/pdweb/www-default/certs directory
4. Start the iKeyman utility on the Reverse Proxy node. Enter the following command on a command window:

```
# gsk7ikm
```

5. From the menu bar, select **Key Database File** →**Open**.
6. In the Open window, complete the following tasks:
   a. For Key database type, select **CMS**.
   b. For Filename, type `pdsrv.kdb`.
   c. For Location, type `/var/pdweb/www-default/certs`.
   d. Click **OK**.
7. When prompted, enter the password for the keystore. By default, the password is `pdsrv`.

   You can change the password by selecting **Key Database File** →**Change Password**.
8. From the Key database content drop-down list, select **Signer Certificates**.
9. Click **Add**.

10. In the Add CA's Certificate from a file window, complete the following tasks:

    a. For Data type, select **Base64-encoded ASCII data**.
    b. For Certificate file name, type `wp_httpd_cert.arm`.
    c. For Location, type `/var/pdweb/www-default/certs`.
    d. Click **OK**.

11. Enter a label when you are prompted. We entered `WP HTTP Server SSL Key`. For consistency, we entered the same name as used when we created the key. Then click **OK**.

    You should see the newly imported certificate listed (for example, WP HTTP Server SSL Key) among the Signer Certificates.

3.5.5 Exporting the WebSEAL certificate

You can create a self-signed certificate for WebSEAL and then export the certificate. Although WebSEAL provides a test certificate, we chose not to use this certificate for our example. This test certificate is included with the distribution of Tivoli Access Manager available to all clients and, therefore, is not secure. For this reason, we create a new self-signed certificate.

Creating a WebSEAL self-signed certificate

To create a self-signed certificate for WebSEAL, use the steps:

1. If you closed the Key Management utility after the previous step, start the Key Management utility and open the key database file (pdsrv.kdb).
2. From the Key database content drop-down list, select **Personal Certificates**.
3. From the menu bar, select **Create →New Self Signed Certificate**.
4. In the Create New Self Signed Certificate window, complete these tasks:

    a. For Key Label, type `WebSEAL default key`. In this case, default is the name of the WebSEAL instance.
    b. For Common Name, type `wslx1.itso.ral.ibm.com`.
    c. For Organization, type `IBM`.
    d. Click **OK**.

5. When you see the message `Do you want to set the key as the default key in the database`, click **No**. In our example, the key is defined explicitly in the webseald-default.conf file.

Exporting the WebSEAL certificate

To export the WebSEAL certificate, follow these steps:

1. If you closed the Key Management utility after the previous step, start the Key Management utility and open the key database file (pdsrv.kdb).

2. Under Personal Certificates, select the certificate that you created in “Creating a WebSEAL self-signed certificate” on page 112 (for example, `WebSEAL default key`).
3. Click **Extract Certificate**.
4. In the Extract Certificate to a File window, complete the following tasks:
   a. For Data Type, select **Base64-encoded ASCII data** (default).
   b. For Certificate file name, type `webseal_default_cert.arm`.
   c. For Location, type `/var/pdweb/www-default/certs`.
   d. Click **OK**.
5. Close the iKeyman utility.

Modifying webseald-default.conf to use a new certificate

In our example, we chose to create a new self-signed certificate. In order for WebSEAL to use this new certificate, we need to modify the webseald-default.conf to define the new key label.

1. Navigate to the /opt/pdweb/etc directory on the Reverse Proxy node (wslx1).
2. Open the webseald-default.conf file with a text editor.
3. Search for webseal-cert-keyfile-label. Modify the value as follows:

   ```
   webseal-cert-keyfile-label = WebSEAL default key
   ```

4. Save and close the file.
5. Restart the WebSEAL instance by running the following command from a console window as a root user:

   ```
   # pdweb restart
   ```

3.5.6 Importing WebSEAL certificate into IBM HTTP Server keystore

To import the WebSEAL certificate into the IBM HTTP Server keystore, complete these tasks:

1. Copy the WebSEAL certificate (for example, `webseal_default_cert.arm`) to the Portal Server node.
   – From Reverse Proxy node (wslx1): /var/pdweb/www-default/certs
   – To Policy Server node (wpslx1): /opt/IBMHttpServer/ssl
2. Start the IBM Key Management Utility on the Portal Server node:

   ```
   # gsk7ikm
   ```

3. From the menu bar, select **Key Database File** →**Open**.
4. Select **/opt/IBMHttpServer/ssl/keyfile.kdb** and click **Open**.
5. When prompted, enter the password for the keystore.

6. From the Key database content drop-down list, select **Signer Certificates**. Click **Add**.
7. In the Add CA's Certificate from a file window, complete the following steps:
   a. For Data type, select **Base64-encoded ASCII data**.
   b. For Certificate file name, type `webseal_default_cert.arm`.
   c. For Location, type `/opt/IBMHttpServer/ssl`.
   d. Click **OK**.
8. When prompted for the label, type the label name. We entered `WebSEAL default key`. For consistency, we entered the same name as used when we created the key. Then click **OK**.

   You should see the newly imported certificate listed (for example, `WebSEAL default key`) among the Signer Certificates.
9. Close the IBM Key Management Utility.

3.5.7 Enabling mutual SSL for the IBM HTTP Server

Enable mutual SSL for the IBM HTTP Server on the Portal Server node as explained in the following steps:

1. Stop the IBM HTTP Server on the Portal Server node:

   ```
   # /opt/IBMHttpServer/bin/apachectl stop
   ```
2. Modify the /opt/IBMHTTPServer/conf/httpd.conf file with a text editor.
   a. Search for the keyword `SSLClientAuth` inside the `<VirtualHost` entry.
   b. Add the following entry after the commented SSLClientAuth entries:

   ```
   SSLClientAuth required
   ```
   c. Save the changes to the httpd.conf file.
3. Start the IBM HTTP Server:

   ```
   # /opt/IBMHttpServer/bin/apachectl start
   ```

   > **Note:** After you set the SSLClientAuth required keyword and value, you are no longer able to directly connect to the IBM HTTP Server via HTTPS. HTTPS connections to the IBM HTTP Server are permitted only from WebSEAL on the Reverse Proxy node.

4. Verify that you are not able to access the IBM HTTP Server directly via HTTPS by entering the following URL in a Web browser:

   ```
   https://wpslx1.itso.ral.ibm.com
   ```

   You should see the message, `Forbidden - You don't have permission to access / on this server.`

Note: When we create a WebSEAL junction in 3.6.3, “Creating a WebSEAL junction” on page 122, we enable mutual SSL for the WebSEAL junction.

3.6 Configuring portal authentication with TAM using TAI

In the ITSO example, our runtime is currently configured for users to directly log in to the WebSphere Portal on the Portal Server node. After completing this section, the authentication for WebSphere Portal is performed by a combination of WebSEAL, on the Reverse Proxy node, and the Tivoli Access Manager Policy Server, on the Policy Server node.

This procedure requires the following tasks:

1. Applying Tivoli Access Manager ACLs to new LDAP suffixes
2. Defining additional MIME types for WebSphere Application Server
3. Creating a WebSEAL junction
4. Enabling forms authentication on WebSEAL
5. Configuring WebSEAL to modify URLs to back-end systems
6. Configuring additional WebSEAL parameters
7. Importing WebSphere Portal users and groups into TAM
8. Defining access controls for WebSphere Portal URIs
9. Configuring the junction mapping table
10. Configuring single signon for WebSEAL and WebSphere via TAI
11. Configuring Portal login and logout for use with WebSEAL

3.6.1 Applying Tivoli Access Manager ACLs to new LDAP suffixes

When Tivoli Access Manager V5.1 is configured, it attempts to apply appropriate access control in the form of access control lists (ACLs) to every LDAP suffix that exists at the time in the LDAP server. In our example, we created the LDAP suffix dc=itso,dc=ibm,dc=com in 3.4.1, “Creating suffixes” on page 92, after we configured Tivoli Access Manager. For this reason, we must apply Tivoli Access Manager ACLs manually to the suffix.

For more information about applying Tivoli Access Manager ACLs to a new LDAP suffix, refer to Appendix D, “Managing user registries in Applying IBM Tivoli Access Manager ACLs to new LDAP suffixes,” in the *Base Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1360.

Note: An alternative solution exists for applying Tivoli Access Manager ACLs to a new LDAP suffix using an LDIF file import in the Tivoli Directory Server. The ITSO sample code includes tam-acls.ldif, which is found in the /tmp/9121code/config/ldap/ directory after it is unpacked. If you choose to import ACLs via the LDIF, you can skip the rest of this section.

To apply Tivoli Access Manager ACLs to a new LDAP suffix, do these steps:

1. Modify the tam-acls.ldif file for your suffix.
2. From a command line, enter:

   ```
   ldapmodify -h ldaplx1.itso.ral.ibm.com -D cn=root -w <password> -i
   /tmp/9121code/config/ldap/tam-acls.ldif
   ```

To apply Tivoli Access Manager ACLs to a new LDAP suffix, perform the following steps on the Policy Server node:

1. Verify that IBM Tivoli Directory Server is started on the Directory Server node.
2. Start server1 via the command prompt:

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh server1
   ```

3. Start the Tivoli Directory Server Web Administration Tool in a Web browser at:

   ```
   http://tamlx1.itso.ral.ibm.com:9080/IDSWebApp/IDSjsp/Login.jsp
   ```

   Here *tamlx1.itso.ral.ibm.com* is the host name of the application server where the IBM Directory Server Web Administration Tool is installed.

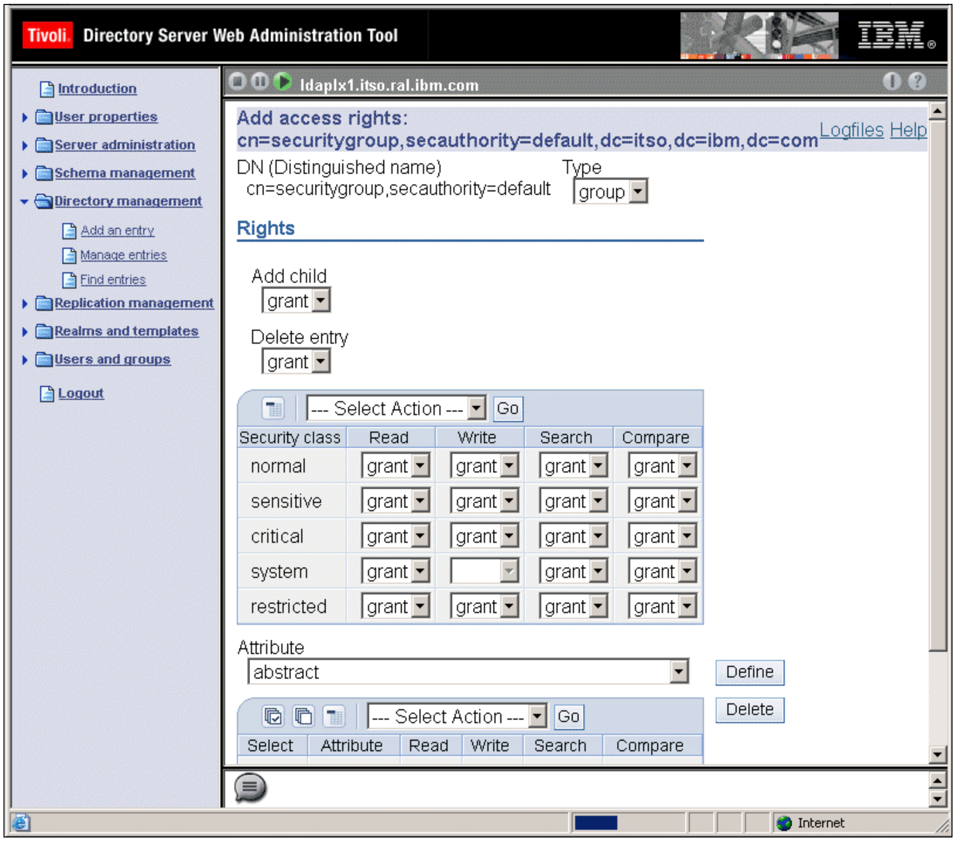
4. From the Web Administration Tool, complete the following tasks:
   a. From the drop-down list on the Login page, select the newly created server (for example, **tamlx1.itso.ral.ibm.com**).
   b. For Username, type `cn=root`.
   c. Enter a password.
   d. Click **Login**.
5. From the Web Administration Tool, select **Directory management** → **Manage entries**.
6. On the Manage entries page, in the Select column, click to select the suffix from the list (for example, dc=itso,dc=ibm,dc=com). Then click **Edit ACL**.
7. To add the cn=SecurityGroup,secAuthority=Default ACL to the ITSO suffix dc=itso,dc=ibm,dc=com, complete the following tasks:
   a. On the Edit ACL page, click **Non-filtered ACLs**.
   b. On the Non-filtered ACLs page, complete the following tasks:
      i. Select **Propagate ACLs**.
      ii. For DN, type: `cn=SecurityGroup,secAuthority=Default`.

iii. For Type, select **group**.
iv. Click **Add**.

c. In the Add access rights page (Figure 3-2), complete these steps:

i. For Add child, select **grant**.
ii. For Delete entry, select **grant**.
iii. Select **grant** for all security classes (normal, sensitive, critical, system, restricted) and actions (read, write, search, compare).
iv. Click **OK** at the bottom of the page.

> **Note:** You cannot grant write permission for the system security class (menu option is disabled).



Add access rights: cn=securitygroup,secauthority=default,dc=itso,dc=ibm,dc=com

DN (Distinguished name): cn=securitygroup,secauthority=default

Type: group

Rights

Add child: grant

Delete entry: grant

Security class	Read	Write	Search	Compare
normal	grant	grant	grant	grant
sensitive	grant	grant	grant	grant
critical	grant	grant	grant	grant
system	grant		grant	grant
restricted	grant	grant	grant	grant

Attribute: abstract

Select	Attribute	Read	Write	Search	Compare

Figure 3-2 Adding an ACL for DN cn=SecurityGroup,secAuthority=Default

8. To add the cn=ivacld-servers,cn=SecurityGroups,secAuthority=Default ACL to the ITSO suffix dc=itso,dc=ibm,dc=com, follow these steps:

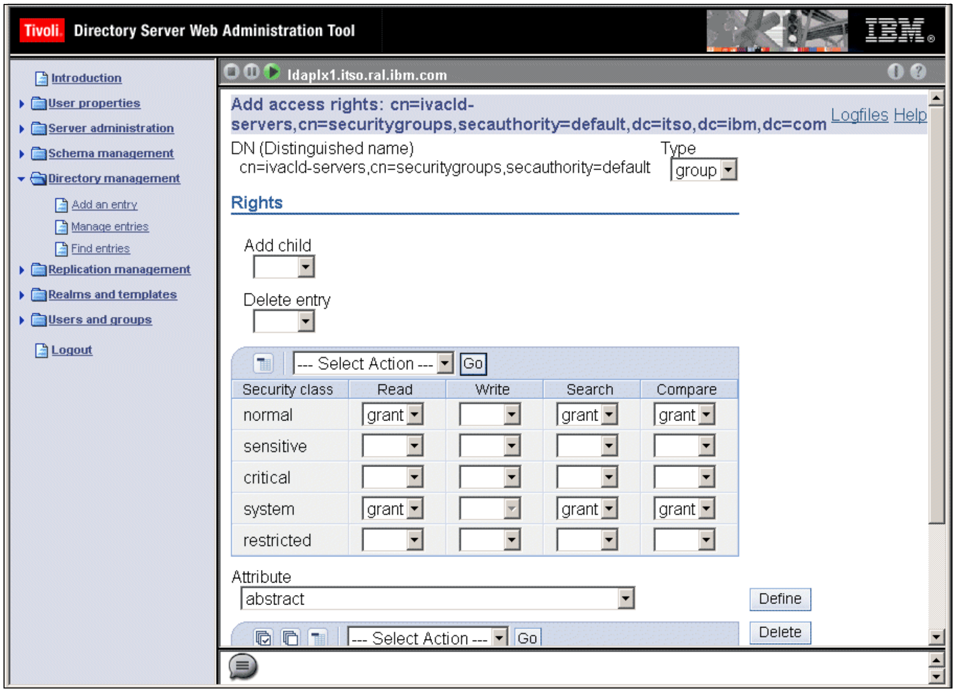
   a. On the Edit ACL page, click **Non-filtered ACLs**.

   b. On the Non-filtered ACLs page, complete the following tasks:

      i. Select **Propagate ACLs**.
      ii. For DN, type: `cn=ivacld-servers,cn=SecurityGroups,secAuthority=Default` (all on one line without any spaces).
      iii. For Type, select **group**.
      iv. Click **Add**.

   c. On the Add access rights page, complete these tasks (see Figure 3-3):

      i. Leave the Add child field blank.
      ii. Leave the Delete entry field blank.
      iii. In the table, for the Security classes *normal* and *system*, select **grant** under the *read*, *search*, and *compare* actions. Leave the Add child, Delete entry, and all other Security classes blank.
      iv. Click **OK** at the bottom of the page.



Add access rights: cn=ivacld-servers,cn=securitygroups,secauthority=default,dc=itso,dc=ibm,dc=com

DN (Distinguished name): cn=ivacld-servers,cn=securitygroups,secauthority=default

Type: group

Rights

Add child:

Delete entry:

--- Select Action --- Go

Security class	Read	Write	Search	Compare
normal	grant		grant	grant
sensitive				
critical				
system	grant		grant	grant
restricted				

Attribute: abstract

Define

--- Select Action --- Go

Delete

Figure 3-3 Add ACL for cn=ivacld-servers,cn=SecurityGroups,secAuthority=Default

9. To add the cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default ACL to the ITSO suffix dc=itso,dc=ibm,dc=com, follow these steps:

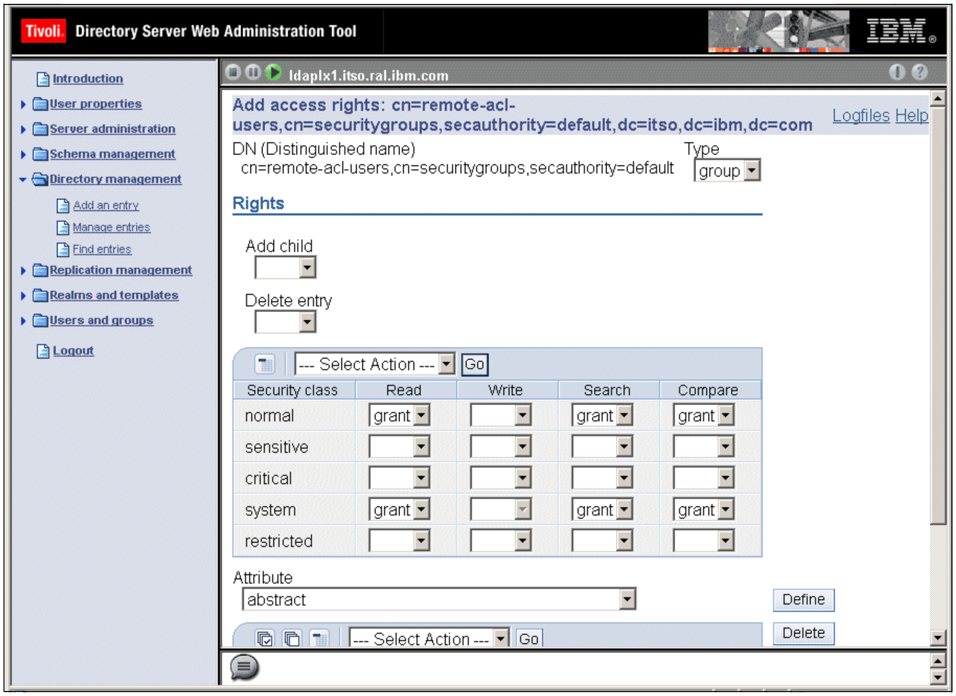
   a. On the Edit ACL page, click **Non-filtered ACLs**.

   b. On the Non-filtered ACLs page, complete the following tasks:

      i. Select **Propagate ACLs**.
      ii. For DN, type `cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default` (all on one line without any spaces).
      iii. For Type, select **group**.
      iv. Click **Add**.

   c. On the Add access rights page (Figure 3-4), complete these tasks:

      i. Leave the Add child field blank.
      ii. Leave the Delete entry field blank.
      iii. In the table, for the Security classes *normal* and *system*, select **grant** under the *read*, *search*, and *compare* actions. Leave the Add child, Delete entry, and all other Security classes blank.
      iv. Click **OK** at the bottom of the page.



Add access rights: cn=remote-acl-users,cn=securitygroups,secauthority=default,dc=itso,dc=ibm,dc=com

DN (Distinguished name): cn=remote-acl-users,cn=securitygroups,secauthority=default

Type: group

Rights

Add child:

Delete entry:

--- Select Action --- Go

Security class	Read	Write	Search	Compare
normal	grant		grant	grant
sensitive				
critical				
system	grant		grant	grant
restricted				

Attribute: abstract

Figure 3-4 ACL for cn=remote-acl-users,cn=SecurityGroups,secAuthority=Default

10. To add the cn=anybody ACL to the ITSO suffix dc=itso,dc=ibm,dc=com, follow these steps:

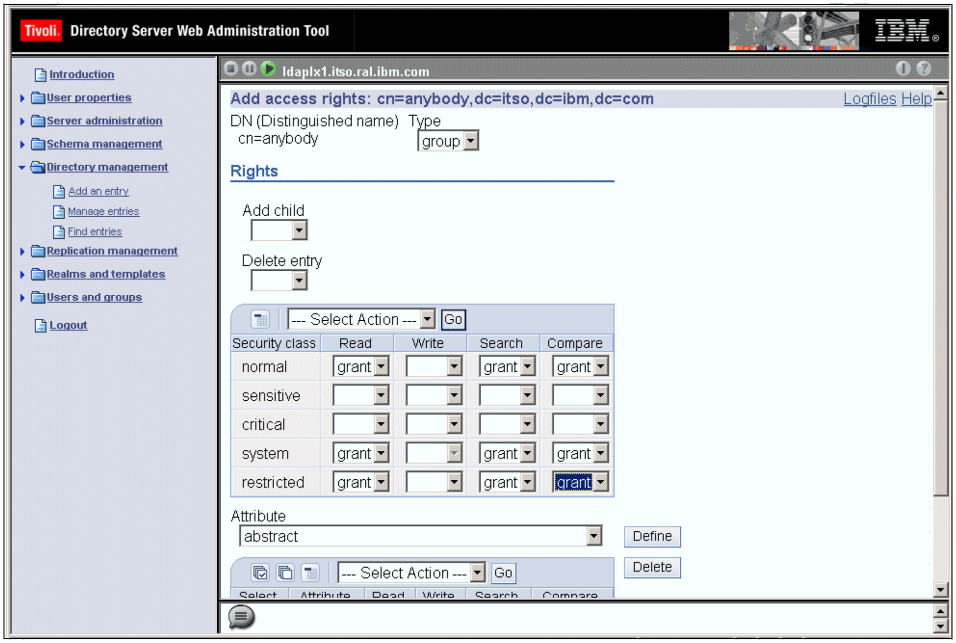
   a. On the Edit ACL page, click **Non-filtered ACLs**.

   b. On the Non-filtered ACLs page, complete the following tasks:

      i. Select **Propagate ACLs**.
      ii. For DN, type `cn=anybody`.
      iii. For Type, select **group**.
      iv. Click **Add**.

   c. On the Add access rights page (Figure 3-5), complete these tasks:

      i. Leave the Add child field blank.
      ii. Leave the Delete entry field blank.
      iii. In the table, for the Security classes *normal*, *system*, and *restricted*, select **grant** under the *read*, *search*, and *compare* actions. Leave the Add child, Delete entry, and all other Security classes blank.
      iv. Click **OK** at the bottom of the page.



Add access rights: cn=anybody,dc=itso,dc=ibm,dc=com

DN (Distinguished name): cn=anybody
Type: group

Rights

Add child:

Delete entry:

Security class	Read	Write	Search	Compare
normal	grant		grant	grant
sensitive				
critical				
system	grant		grant	grant
restricted	grant		grant	grant

Attribute: abstract

Select	Attribute	Read	Write	Search	Compare

Figure 3-5 Adding an ACL for cn=anybody

11. After you finish adding the ACLs, click **OK**.

12. On the Manage entries page, click **Close**.

    The LDAP server does not need to be restarted for the changes to take effect.

13. If you are finished with the IBM Directory Server Web Administration Tool, click **Logout**.

3.6.2 Defining additional MIME types for WebSphere Application Server

By default WebSphere Application Server V5 is not configured with MIME types for Java Archive (JAR) files and Microsoft ActiveX Control files. These MIME types are commonly used by back-end Web applications such as Lotus® components included in IBM WebSphere Portal Extend for Multiplatforms V5.0.2. When using Tivoli Access Manager WebSEAL, the MIME type must be defined in response headers in order for the response to be passed through WebSEAL.

Table 3-4 lists the MIME type definitions that we add in this section. If your portlet application uses other MIME types that are not found by default within WebSphere Application Server, follow the same procedure to add the MIME type definitions.

Table 3-4 Additional MIME types for WebSphere Portal

Description	MIME type	Extensions
Java Archive	application/java-archive	jar
ActiveX Control	application/x-cabinets-Win32-x86	cab

To add MIME type definitions to the WebSphere Application Server where WebSphere Portal is installed, use the following steps:

1. Ensure that the server1 application server is started on the Portal Server node.
2. Start the WebSphere Application Server Administrative Console:

   `https://wpslx1.itso.ral.ibm.com:9043/admin`

3. Log on as the WebSphere Application Server administrator user ID and password (for example, `wpsbind`).
4. Select **Environment** →**Virtual Hosts**.
5. On the Virtual Hosts page, click **default_host**.
6. Add a new MIME type.
   a. Under Additional Properties, click **MIME Types**.
   b. Click **New**.

c. On the New MIME Type page, enter the following values for the Java archive, as shown in Table 3-4:

- MIME type: `application/java-archive`
- Extensions: `jar`

Then click **OK**.

7. Repeat step 6 to add a new type for ActiveX controls with the values listed in Table 3-4.
8. Click **Save**.
9. On the Save to Master Configuration page, click **Save**.
10. Click **Logout**.

3.6.3 Creating a WebSEAL junction

To create the WebSEAL junctions for our configuration, we use the Tivoli Access Manager **`pdadmin`** command line interface. You can use the **`pdadmin`** command line interface in one of the following three modes:

- Single command mode
- Interactive command mode
- Multiple command mode

For the ITSO example, we chose to create an input file. In our example, we use the multiple command mode with a command file that contains the commands to create the junction. Figure 3-6 shows the ITSO sample /portal WebSEAL junction for WebSphere Portal.

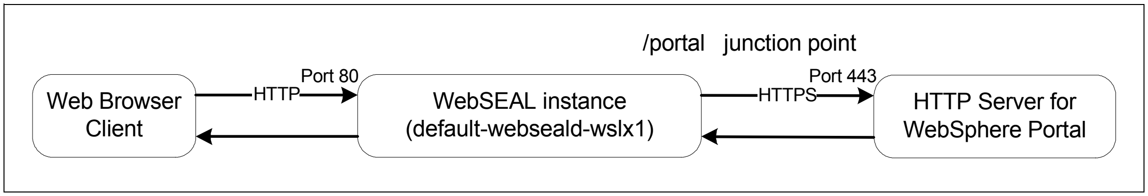

/portal junction point
Web Browser Client
HTTP
Port 80
WebSEAL instance (default-webseald-wslx1)
HTTPS
Port 443
HTTP Server for WebSphere Portal

Figure 3-6 ITSO junction for WebSphere Portal

Creating the wp-junction.pd command file

For our example, we created a command file named wp-junction.pd. This file is included in the /tmp/9121code/config/tam directory after unzipping the ITSO sample code.

To create the wp-junction.pd file that contains the commands to create a junction for WebSphere Portal, follow this procedure:

1. Ensure that the Access Manager Policy Server Windows service is started.
2. Start and log on to the **pdadmin** command line interface on the Reverse Proxy node, by entering the following command in a console window:

   ```
   # cd /opt/PolicyDirector/bin
   # pdadmin -a sec_master
   ```

3. When prompted, enter the sec_master and password.
4. To get a list of servers, enter the following command:

   ```
   > server list
   ```

   The command outputs list that contains the following result:

   ```
   default-webseald-wslx1
   ```

   Note the WebSEAL server name.

5. We created the wp-junction.pd file with the commands as in Example 3-5. The wp-juncion.pd file is used to define an SSL junction for WebSphere Portal in the WebSEAL instance that we configured previously. This enables mutual SSLs for the WebSEAL junction that was created.

   The syntax of this command is:

   ```
   server task webseal_servername create -t junction_type -h
   backend_server_hostname -p backend_server_port -j -w -c all -K “key_label”
   -D “CN” junction_point
   ```

Example 3-5 ITSO wp-junction.pd sample command file

```
server task default-webseald-wslx1 create -t ssl -h wpslx1.itso.ral.ibm.com -p 443 -j -w -c all
-K "WebSEAL default key" -D "CN=wplx1.itso.ral.ibm.com,O=IBM,C=US" /portal
```

> **Note:** For scenarios using the internal HTTP service built-in to WebSphere Application Server (development WebSphere Test Environment or no external Web server), update the port to 9444 for SSL instead of port 443 in the wp-junction.pd (see Example 3-5).

The parameters are as follows:

- *webseal_servername* is the server name returned from the server list command in the previous step (for example, default-webseald-wslx1).
- *junction_type* is either tcp for non-ssl or ssl.
- *backend_server_hostname*, in our example, is Portal Server node.
- *backend_server_port*, in our example, is Portal Server node port 443.

- -j enables junction cookies for handling server relative URLs.
- -w is used to support the Win32 file system. This option allows only full paths to resources (not the short name). Also, it makes the resource validation case insensitive.
- -c all is one of two key options for WebSphere Portal integration. The other option is -c iv-user. When performing authentication only, -c -iv-user is sufficient. When using performing authentication and authorization, then use -c all.
- -K *key_label* is the *WebSEAL key label* on the Reverse Proxy node.
- -D *CN*, in our example, is `CN=wpslx1.itso.ral.ibm.com,O=IBM,C=US`.

> **Note:** For details about the parameters, refer to Appendix B, "WebSEAL junction reference," in the *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359.

Creating the WebSEAL junction from wp-junction.pd

To create the WebSEAL junction from wp-junction.pd, follow theses steps:

1. Ensure that the following servers are started:
   - Directory Server node
     - Tivoli Directory Server
   - Policy Server node
     - Policy Server
     - Authorization Server
   - Portal Server node
     - IBM HTTP Server
     - WebSphere Portal Server
   - Reverse Proxy node
     - WebSEAL
2. Copy the wp-junction.pd command file to the /tmp directory on the Reverse Proxy node.
3. Open a console command window (not pdadmin).
4. From the command line, change to the directory of the wp-junction.pd file and enter the following command:

   ```
   > pdadmin -a sec_master wp-junction.pd
   ```
5. When prompted for the password, enter the `sec_master` password.

Note: When creating the junction, WebSEAL attempts to connect to the back-end system (for example, WebSphere Portal). If the system is not available, you see the following message:

```
DPWWA1222E   A third-party server is not responding. Possible causes: the
server is down, there is a hung application on the server, or network
problems. This is not a problem with the WebSEAL server.
DPWIV1054E   Could not connect
```

The junction is still created.

6. Verify that the junction was created properly.
   a. Start `pdadmin` and log in as `sec_master`.

      ```
      # pdadmin -a sec_master
      ```

   b. Enter the following command to verify the junction:

      ```
      server task default-webseald-wslx1 list
      ```

      You should see a list of junctions including the new junction created, for example /portal.

   c. You can further review the junction definition by entering the following command. This lists all settings for the /portal junction.

      ```
      server task default-webseald-wslx1 show /portal
      ```

3.6.4 Enabling forms authentication on WebSEAL

By default, WebSEAL uses HTTP basic authentication for its authentication challenge to the user. With basic authentication, the logout does not happen until the user closes the Web browser. In addition, there are other security issues with basic authentication that make forms authentication preferable.

To enable forms authentication, you must update webseald-*webseal_instancename*.conf on the Reverse Proxy node as follows:

1. Navigate to the following directory:

   ```
   # cd /opt/pdweb/etc
   ```

2. Back up the webseald-default.conf (for example, webseald-default.conf.org).

3. We modified the webseald-default.conf, as shown in Example 3-6, for the ITSO example.

Example 3-6 ITSO example forms authentication settings in webseald-default.conf

```
[ba]
#-------------------------
# BASIC AUTHENTICATION
#-------------------------
#Enable authentication using basic authentication mechanism
#One of <http, https, both, none>
ba-auth = none


[forms]
#-------------------------
#Forms
#-------------------------
#Enable authentication using forms
#One of <http, https, both, none>
forms-auth = https
```

4. Save the webseald-default.conf file.
5. Restart the Access Manager WebSEAL to enable these configuration changes:

   ```
   # pdweb restart
   ```

6. To verify that the forms authentication is enabled, access WebSEAL with Web browser:

   ```
   https://wslx1.itso.ral.ibm.com
   ```

7. You should now see the WebSEAL login form page in the browser instead of the WebSEAL basic authentication pop-up. Log on to WebSEAL with the `sec_master` user ID and password.

   You should see the WebSEAL splash screen.

Now that forms are enabled, users can log out without needing to close the Web browser.

3.6.5 Configuring WebSEAL to modify URLs to back-end systems

Web pages returned to the client from back-end applications are likely to contain URL links to resources located on those application servers (for example, WebSphere Application Server or WebSphere Portal). It is important that these links are constructed to direct any requests back to the correct locations of these resources.

This section explains how to configure WebSEAL to modify URLs to back-end systems for:

- Enabling WebSEAL URL filtering
- Enabling WebSEAL processing of URLs in the request

Enabling WebSEAL URL filtering

The Tivoli Access Manager WebSEAL URL filtering performs two functions:

- It adds the junction name to the path of the absolute and server-relative URLs that refer to resources located on the back-end servers.
- The absolute URL host/port is mapped to the respective junction on WebSEAL.

When adding WebSEAL in front of a back-end application, such as WebSphere Application Server or WebSphere Portal, the absolute URLs of the back-end application need to be mapped to WebSEAL.

> **Note:** For more information about WebSEAL URL filtering, refer to the "Filtering URLs in responses" section in Chapter 10 in the *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359.

If the back-end application (WebSphere Portal) generates an absolute URL (see Table 3-5), it must be mapped on WebSEAL since only the Reverse Proxy node is accessible to Web browser clients. In the filtered WebSEAL URL, the junction /portal was created in 3.6.3, "Creating a WebSEAL junction" on page 122.

Table 3-5 WebSEAL URL filter example

WebSphere Portal URL	Filtered WebSEAL URL
http://wpslx1.itso.ral.ibm.com/wps/portal	http://wslx1.itso.ral.ibm.com/portal/wps/portal

To enable WebSEAL URL filtering, complete these steps:

1. Navigate to the following directory:

   ```
   # cd /opt/pdweb/etc
   ```

2. Back up the webseald-default.conf (for example, webseald-default.conf.org).
3. Modify the webseald-default.conf, as shown in Example 3-7, for the ITSO example to enable WebSEAL URL filtering.

Example 3-7 ITSO example URL filtering settings in webseald-default.conf

```
[script-filtering]
script-filter = yes
```

4. Save the webseald-default.conf file.
5. Restart the Access Manager WebSEAL-default Windows service for the changes to take effect. We make other changes to the webseald-default.conf file, so you may defer the restart at this time.

Enabling WebSEAL processing of URLs in the request

When URLs are dynamically generated by client-side applications, such as applets, or embedded scripts in the HTML, such as JavaScript or ActiveX, the URLs need to be mapped. In this case, WebSEAL does not have the ability to apply its standard filtering rules to the dynamically generated URLs.

WebSEAL provides two methods of addressing dynamically generated URLs from client-side applications:

- *Junction cookies* are specified with a -j option when creating the junction.
- *Junction mapping* uses a static junction mapping table to map specific back-end resources to junction names.

> **Note:** For more information about WebSEAL URL filtering, refer to the “Processing URLs in requests” section in Chapter 10 in the *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359.

For the ITSO example, we use the junction cookies option for the base configuration. When creating the junction in 3.6.3, “Creating a WebSEAL junction” on page 122, we use the -j option for junction cookies of processing URLs in the request.

3.6.6 Configuring additional WebSEAL parameters

This section describes the additional parameters we modified in the webseald-default.conf file for the ITSO working example.

> **Note:** For information about configuring WebSEAL parameters, refer to Appendix A, “WebSEAL configuration file reference,” in the *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359.

To configure additional WebSEAL parameters, such as timeout, follow these steps:

1. Navigate to the following directory:

```
# cd /opt/pdweb/etc
```

2. Modify the webseald-default.conf as shown in Example 3-8 to configure additional parameters. The webseald-default.conf includes comments that explain each parameter. The ITSO example contains additional parameters in webseald-default.conf.

Example 3-8 Additional parameters for the ITSO webseald-default.conf

```
[server]
dynurl-allow-large-posts=yes

[junction]
http-timeout=300
https-timeout=300

[session]
ssl-id-sessions=no
```

3. Save the webseald-default.conf file.
4. Restart the Access Manager WebSEAL-default Windows service for the changes to take effect:

   ```
   # pdweb restart
   ```

3.6.7 Importing WebSphere Portal users and groups into TAM

Although the users and groups used by WebSphere Portal are already created in the Tivoli Directory Server LDAP directory, the users have not been imported into the Tivoli Access Manager. The importing of users into Tivoli Access Manager includes adding attributes to existing users in the LDAP directory.

Creating the wp-tam-user-import.pd command file

In our example, we create a command file called wp-tam-user-import.pd found in the ITSO sample code /tmp/9121code/config/tam directory. This command file is used to import the WebSphere Portal users into Tivoli Access Manager using pdadmin. Example 3-9 shows the ITSO-provided wp-tam-user-import.pd file.

Example 3-9 ITSO example wp-tam-user-import.pd file

```
user import -gsouser wpsadmin uid=wpsadmin,cn=users,dc=itso,dc=ibm,dc=com
user modify wpsadmin account-valid yes
user modify wpsadmin password-valid yes
group import wpsadmins cn=wpsadmins,cn=groups,dc=itso,dc=ibm,dc=com
```

Importing WebSphere Portal users into TAM via command file

To import the WebSphere Portal users and groups into Tivoli Access Manager using a **pdadmin** command file, follow these steps:

1. Open a console window as a root user.
2. Copy the wp-tam-user-import.pd file to a temporary directory on the Reverse Proxy node (for example, /tmp).
3. From the temporary directory, enter the following command to import the WebSphere Portal users and groups into Tivoli Access Manager:

   ```
   pdadmin -a sec_master wp-tam-user-import.pd
   ```

   When prompted, enter the `sec_master` password.
4. To verify that the users and groups where imported properly, enter the following commands from the **`pdadmin`**:

   ```
   user list * 100
   group list * 100
   ```

3.6.8 Defining access controls for WebSphere Portal URIs

There are four access categories to define for WebSphere Portal, as shown in Table 3-6.

Table 3-6 Access categories for WebSphere Portal

Access category	Description
WP_all_access	Access for all users (authenticated and unauthenticated)
WP_authenticated_access	Access for authenticated users only
WP_admin_access	Access for administrator users only
WP_no_access	No access

Creating TAM objects for WebSphere Portal URIs

To create Tivoli Access Manager objects for WebSphere Portal URIs, perform the following steps on the Reverse Proxy node:

> **Note:** For more information about creating Tivoli Access Manager objects, refer to Chapter 12, “Application integration,” in the *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359.

1. Create a file named dynurl.conf in the /opt/pdweb/www-default/lib directory, as shown in Example 3-10.

   Here the default is the name of the WebSEAL instance. We include a dynurl.conf file in the /tmp/9121code/config/tam ITSO sample code directory.

Example 3-10 ITSO example dynurl.conf

```
/portal/wps/portal     /portal/wps/portal*
/portal/wps/myportal   /portal/wps/myportal*
/portal/wps/config     /portal/wps/config*
/portal/wps/doc        /portal/wps/doc*
/portal/wps            /wps/*
```

> **Note:** We add entries both *before* the mapping and *after* the mapping versions of URLs that will be handled by the JMT. This is required because WebSEAL performs two ACL checks: one on the URL before the JMT transformation and one after. Both must pass for access to be granted.
>
> Considering that the real access control check is the second one, we add dummy entries for the *before* versions and mapped them to an object that is readable by both unauthenticated and authenticated users (/portal/wps), so the first ACL check now always passes.

2. Ensure that the owner ID and group for the dynurl.conf are set to ivmgr:

   ```
   # chown ivmgr:ivmgr /opt/pdweb/www-default/lib/dynurl.conf
   ```

3. To activate the definitions, choose one of the following options:

   – Enter the following command:

   ```
   pdadmin -a sec_master -p <password> server task default-webseald-wslx1
   dynurl update
   ```

   – Restart the Tivoli Access Manager WebSEAL:

   ```
   # pdweb restart
   ```

Defining access control policy for WebSphere Portal

This section explains how to define the access control policy for WebSphere Portal. It includes the following operations:

- Creating the Tivoli Access Manager ACL templates that correspond to the access categories defined for WebSphere Portal
- Updating the ACLs for the imported users and groups
- Attaching ACLs to protected objects for WebSphere Portal

To define the access control policy, follow these steps:

1. Create a command file called wp-tam-acl.pd, as shown in Example 3-11. The file is included in the /tmp/9121code/config/tam ITSO sample code directory.

Note: The ITSO provided wp-tam-acl.pd command file includes three sections. The first two sections in Example 3-11 do not need to be modified unless you need to create new access categories.

The last section in Example 3-11 includes settings that are changed based on your environment. For example, you need to update /WebSEAL/host-instance_name, which represents the beginning of the Web space for a particular WebSEAL server instance (for example, /WebSEAL/wslx1-default). To retrieve your setting, enter the following command on the Reverse Proxy node:

```
pdadmin -a sec_master -p <password> object list /WebSEAL
```

Record the value returned and update the command file accordingly.

Example 3-11 ITSO example wp-tam-acl.pd

```
acl create WP_all_access
acl create WP_authenticated_access
acl create WP_admin_access
acl create WP_no_access

acl modify WP_admin_access set user sec_master TcmdbsvaBrxl
acl modify WP_admin_access set group iv-admin Tcmdbsvarxl
acl modify WP_admin_access set group webseal-servers Tgmdbsrxl
acl modify WP_admin_access set group wpsadmins Tr
acl modify WP_admin_access set any-other T
acl modify WP_admin_access set unauthenticated T
acl modify WP_no_access set user sec_master TcmdbsvaBrxl
acl modify WP_no_access set group iv-admin Tcmdbsvarxl
acl modify WP_no_access set group webseal-servers Tgmdbsrxl
acl modify WP_no_access set group wpsadmins T
acl modify WP_no_access set any-other T
acl modify WP_no_access set unauthenticated T
acl modify WP_authenticated_access set user sec_master TcmdbsvaBrxl
acl modify WP_authenticated_access set group iv-admin Tcmdbsvarxl
acl modify WP_authenticated_access set group webseal-servers Tgmdbsrxl
acl modify WP_authenticated_access set group wpsadmins Tr
acl modify WP_authenticated_access set any-other Tr
acl modify WP_authenticated_access set unauthenticated T
acl modify WP_all_access set user sec_master TcmdbsvaBrxl
acl modify WP_all_access set group iv-admin Tcmdbsvarxl
acl modify WP_all_access set group webseal-servers Tgmdbsrxl
acl modify WP_all_access set group wpsadmins Tr
acl modify WP_all_access set any-other Tr
acl modify WP_all_access set unauthenticated Tr
acl attach /WebSEAL/wslx1-default/portal/wps/config WP_admin_access
acl attach /WebSEAL/wslx1-default/portal/wps/myportal WP_authenticated_access
acl attach /WebSEAL/wslx1-default/portal/wps/portal WP_all_access
acl attach /WebSEAL/wslx1-default/portal/wps/doc WP_all_access
acl attach /WebSEAL/wslx1-default/portal/wps WP_all_access
```

2. To create the Tivoli Access Manager ACLs, modify the access and attach to objects using the wp-tam-acl.pd file by entering:

   ```
   pdadmin -a sec_master -p <password> wp-tam-acl.pd
   ```

3.6.9 Configuring the junction mapping table

Several portlets, including the Resource Permissions portlet, and the productivity component editors use relative JavaScript within the portlet or component. These portlets and components do not function correctly when accessed through a WebSEAL junction. For the JavaScript to be interpreted and navigation followed correctly, WebSEAL must be configured to insert the junction point into the JavaScript. One way to accomplish this is through the use of the JMT table function in WebSEAL.

To enable the JMT function, define an ASCII text file called *jmt.conf* as follows on the Reverse Proxy node:

1. Create the jmt.conf file in the /opt/pdweb/www-default/lib directory.

   The location of this file is specified in the `[junction]` stanza of the webseald-default.conf configuration file jmt-map = lib/jmt.conf.

2. The format for data entry in the table consists of the junction name, a space, and the resource location pattern. You can also use wildcard characters to express the resource location pattern.

   The ITSO provided /tmp/9121code/config/tam/jmt.conf contains the following contents:

   ```
   /portal /wps/*
   ```

   Save and close the file.

3. Set the owner ID and group for jmt.conf to ivmgr for both user and group:

   ```
   # chown ivmgr:ivmgr /opt/pdweb/www-default/lib/jmt.conf
   ```

4. Reload the JMT in WebSEAL:

   a. Open a command window.

   b. Enter the following command to login to **pdadmin**:

      ```
      pdadmin -a sec_master -p <password>
      ```

c. Enter the following command from the **pdadmin** command line to reload the JMT:

```
server task default-webseald-wslx1 jmt load
```

You should see the message `JMT table successfully loaded`.

d. Exit from **pdadmin** by typing `exit`.

e. Restart WebSEAL:

```
pdweb restart
```

5. To verify that the JMT (jmt.conf) is working properly, enter the following URL in a Web browser to access WebSphere Portal via WebSEAL (Reverse Proxy node host name):

```
http://wslx1.itso.ral.ibm.com/wps/portal
```

You should see the default WebSphere Portal page for unauthenticated users.

3.6.10 Configuring single signon for WebSEAL and WebSphere via TAI

When using a Reverse Proxy, such as WebSEAL, to authenticate users in the demilitarized zone (DMZ), WebSphere Application Server, as well as other back-end applications and services, should trust the authentication that has been performed and the identity that is presented by the Reverse Proxy. If this trust is established, users need only to authenticate once to the Reverse Proxy in order to access all authorized services located beyond that proxy. This is commonly known as *Reverse Proxy Single Signon* (RPSS).

> **Note:** For further explanation, refer to the “Design and integration guidelines” chapter in *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

There are two ways to establish a trust relationship between WebSphere Application Server and WebSEAL:

- Trust Association Interceptor (TAI)

  For the ITSO working example runtime environment, we use TAI for the implementation procedure found in this section.

- Lightweight Third Party Authentication (LTPA) Token

  For details about implementing using an LTPA Token for the single signon (SSO) configuration, refer to the “Configure single signon using LTPA” appendix in *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

There are three possible methods to verify that the request to the WebSphere Application Server came from WebSEAL:

- TCP junction without SSL, with basic authentication credentials supplied
- SSL junction with basic authentication
- Mutual SSL junction without basic authentication credentials

> **Note:** We used this method for the ITSO example.

Enabling TAI on the Portal Server node

To enable TAI in WebSphere Application Server on the Portal Server node using the WebSphere Application Server Administrative Console, follow these steps:

1. Ensure that the WebSphere Application Server server1 is started on the Portal Server node:

   ```
   # /opt/WebSphere/AppServer/bin/serverStatus.sh -all -username wpsbind
   -password wpsbind
   ```

   If server1 is not started, start the server as follows:

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh server1
   ```

2. Start the WebSphere Application Server Administrative Console:

   a. Enter the URL:

   ```
   https://wpslx1.itso.ral.ibm.com:9043/admin
   ```

   b. Enter WebSphere administrator credentials (for example, wpsbind).

3. Select **Security** →**Authentication Mechanisms** →**LTPA**.

   > **Note:** Although LTPA is the menu option, we are configuring TAI.

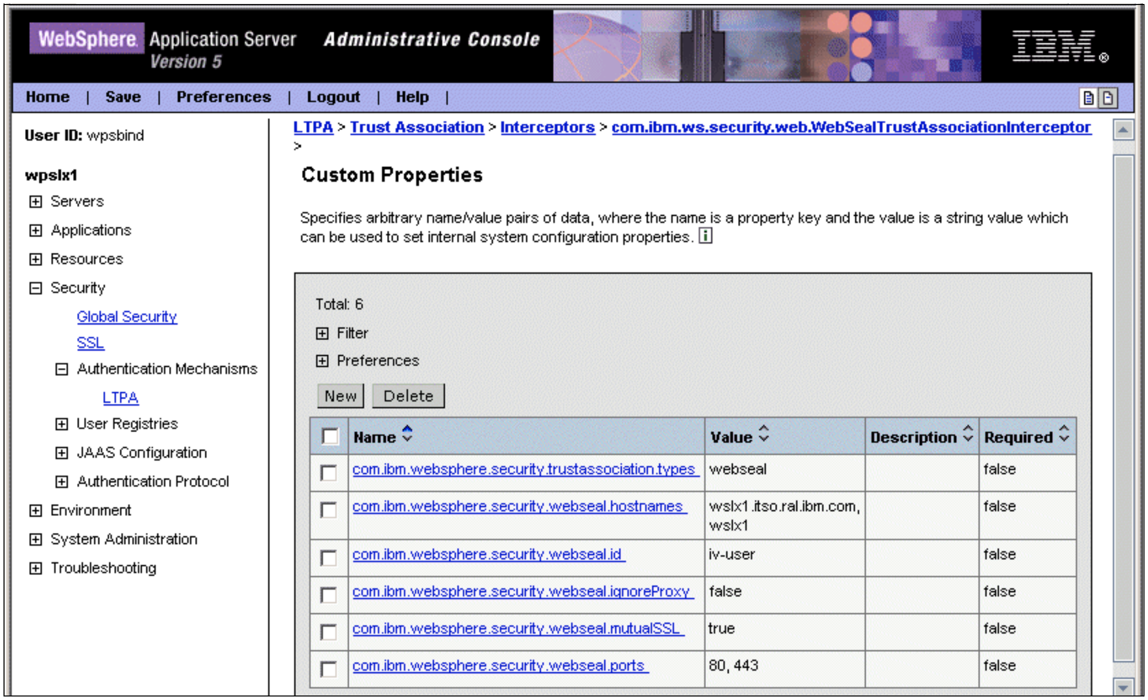
4. Under Additional Properties, click **Trust Association**.
5. On the Trust Association page, select the **Trust Association Enabled** check box. Click **Apply**.
6. Under Additional Properties, click **Interceptors**.
7. Click **com.ibm.ws.security.web.WebSealTrustAssociationInterceptor**.
8. Under Additional Properties, click **Custom Properties**.
9. In the Custom Properties panel (Figure 3-7), click **New** and enter the General Properties name and value pairs specified in Table 3-7. Add each of the names listed in the table. For more information about the possible values that these properties might have, refer to the WebSphere Portal InfoCenter at:

   http://www.ibm.com/websphere/portal/library

Table 3-7 Custom properties for WebSEAL Trust Association Interceptor

Name	Value
com.ibm.websphere.security.trustassociation.types	webseal
com.ibm.websphere.security.webseal.id	iv-user
com.ibm.websphere.security.webseal.hostnames	wslx1.itso.ral.ibm.com, wslx1 **Note**: This is the Reverse Proxy node in our example. The host name is case sensitive.
com.ibm.websphere.security.webseal.ports	80,443
com.ibm.websphere.security.webseal.ignoreProxy	false
com.ibm.websphere.security.webseal.mutualSSL	true **Note**: SSL between WebSEAL and the IBM HTTP Server on the Portal Server node.

After you create all of the properties, the Custom Properties should appear as shown in Figure 3-7.



LTPA > Trust Association > Interceptors > com.ibm.ws.security.web.WebSealTrustAssociationInterceptor >

Custom Properties

Specifies arbitrary name/value pairs of data, where the name is a property key and the value is a string value which can be used to set internal system configuration properties.

Total: 6

□	Name	Value	Description	Required
□	com.ibm.websphere.security.trustassociation.types	webseal		false
□	com.ibm.websphere.security.webseal.hostnames	wslx1.itso.ral.ibm.com, wslx1		false
□	com.ibm.websphere.security.webseal.id	iv-user		false
□	com.ibm.websphere.security.webseal.ignoreProxy	false		false
□	com.ibm.websphere.security.webseal.mutualSSL	true		false
□	com.ibm.websphere.security.webseal.ports	80, 443		false

Figure 3-7 Trust association properties

Tip: Check and double-check the names *and* values of all the Custom properties before you save the configuration changes.

10. Click **Save**.
11. On the Save to Master Configuration page, click **Save**.
12. Click **Logout**.
13. Restart the WebSphere_Portal application server.

Verifying the TAI configuration

Now that we enabled TAI within WebSphere Application Server on the Portal Server node, we recommend that you verify that TAI is working properly.

1. To verify that access to unauthenticated portal pages is working properly, enter the following URL in a Web browser:

   ```
   http://wslx1.itso.ral.ibm.com/portal/wps/portal
   ```

   In this case, there should be no authentication.

2. To verify that authenticated access is working properly, enter the following URL in a Web browser:

   ```
   https://wslx1.itso.ral.ibm.com/portal/wps/myportal
   ```

   WebSEAL should challenge you to authenticate. After you log in as wpsadmin, you should be directed to the user's secure and personalized myportal page. If this works, you have successfully configured the environment for SSO using TAI.

Note: The Logout link does not work at this stage. In the next section, we configure the WebSphere Portal login/logout for use with WebSEAL.

If you are directed to the portal login window at wps/portal/.scr/Login or the public page, there is a problem with the Trust Association Interceptor configuration.

3.6.11 Configuring Portal login and logout for use with WebSEAL

In our example, we have configured WebSEAL to authenticate users for WebSphere Portal. With our current configuration, it is no longer possible to log in or log out of WebSphere Portal directly. This section explains how to configure the WebSphere Portal login/logout functionality for use with WebSEAL.

Modifying web.xml

To modify the WebSphere Portal login, so that login requests are directed to the personalized portal URL, use these steps:

1. Navigate to the following directory on the Portal Server node:

   ```
   # cd /opt/WebSphere/AppServer/installedApps/wpslx1/wps.ear/wps.war/WEB-INF/
   ```

2. Back up the existing web.xml to web.xml.org.
3. Modify the web.xml contents as shown in Example 3-12.

Example 3-12 ITSO modified web.xml snippet

```
<login-config id="LoginConfig_1">
<auth-method>FORM</auth-method>
<realm-name>WPS</realm-name>
<form-login-config id="FormLoginConfig_1">
<form-login-page>/myportal</form-login-page>
<form-error-page>/error.html</form-error-page>
</form-login-config>
</login-config>
```

4. Save and close the file.

Creating wpslogout.html

When a WebSphere Portal user log outs of WebSEAL, we want to display the public (unauthenticated) portal page at:

```
http://wslx1.itso.ibm.com/portal/wps/portal
```

To achieve this behavior, create the wpslogout.html to redirect WebSEAL logout to the WebSphere Portal public page as follows:

> **Note:** The ITSO sample code /tmp/9121code/config/tam directory includes a sample wpslogout.html file. If you choose to copy the ITSO sample wpslogout.html to the /opt/pdweb/www-default/lib/html/C directory, change the owner of the file as follows:
>
> ```
> # chown ivmgr:ivmgr /opt/pdweb/www-default/lib/html/C/wpslogout.html
> ```

1. Navigate to the following directory on the Reverse Proxy node:

   ```
   # cd /opt/pdweb/www-default/lib/html/C
   ```

2. Create the wpslogout.html file as shown in Example 3-13. Modify the href value to include the WebSEAL host name, for example:

   ```
   http://wslx1.itso.ral.ibm.com/portal/wps/portal
   ```

3. Save and close the file.

Example 3-13 ITSO example wpslogout.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<script language=javascript type="text/javascript">
<!--

// Set this variable to a semi-colon list of the names of cookies
//   you do not want to delete
var exception_list = "";

function delete_cookie (name, path)
{
    // Set expiration date to last year
    var expiration_date = new Date ();
    expiration_date . setYear (expiration_date . getYear () - 1);
    expiration_date = expiration_date . toGMTString ();

    // Expire the cookie
    var cookie_string = name + "=; expires=" + expiration_date;
    if (path != null)
        cookie_string += "; path=" + path;
    document . cookie = cookie_string;
}

function name_in_list (n, lst)
{
    var arr = lst . split ("; ");
    for (var j = 0; j < arr . length; j ++) {
        if (arr[j] == n)
            return true;
    }
    return false;
}

function delete_all_cookies (path, exceptions)
{
    // Get cookie list and split into an array of cookie entries
    var cookie_string = "" + document . cookie;
    var cookie_array = cookie_string . split ("; ");

    // Delete each cookie ...
    //   EXCEPT those whose naems appear in the semicolon delimited list
    //   passed in as the second parameter to this function
    for (var i = 0; i < cookie_array . length; ++ i) {
        var single_cookie = cookie_array [i] . split ("=");
        var name = single_cookie [0];
         if (name_in_list(name, exceptions) == false)

             delete_cookie (name, path);
    }
}

// -->
</script>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="Refresh" content="2;URL=http://wslx1.itso.ral.ibm.com/portal/wps/portal">

<title>PKMS Administration: User Log Out</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" onLoad=delete_all_cookies("/",exception_list)>
<font size="+2"><b>User %USERNAME% has logged out.</b></font>


<BR><BR>
<BR><BR>
Redirecting to public portal page ... select <a
href="http://wslx1.itso.ral.ibm.com/portal/wps/portal">here</a> if your browser does not
automatically redirect after 2 seconds.
</body>
</html>
```

Modifying logout.html

When users from other applications (non WebSphere Portal) accessed through WebSEAL perform log out, the logout.html page is displayed.

> **Note:** The ITSO sample code /tmp/9121code/config/tam directory includes a sample logout.html file. If you choose to copy the ITSO sample logout.html to the /opt/pdweb/www-default/lib/html/C directory, change the owner of the file as follows:
>
> ```
> # chown ivmgr:ivmgr /opt/pdweb/www-default/lib/html/C/logout.html
> ```

1. Navigate to the following directory on the Reverse Proxy node:

   ```
   # cd /opt/pdweb/www-default/lib/html/C
   ```

2. Back up the existing logout.html file to the logout.html.org file.
3. Replace the existing logout.html file contents as shown in Example 3-14.
4. Save and close the file.

Example 3-14 ITSO example logout.html

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<!-- Copyright (C) 2000 Tivoli Systems, Inc. -->
<!-- Copyright (C) 1999 IBM Corporation -->
<!-- Copyright (C) 1998 Dascom, Inc. -->
<!-- All Rights Reserved. -->

<script language=javascript type="text/javascript">
<!--

// Set this variable to a semi-colon list of the names of cookies
//   you do not want to delete
var exception_list = "";

function delete_cookie (name, path)
{
    // Set expiration date to last year
    var expiration_date = new Date ();
    expiration_date . setYear (expiration_date . getYear () - 1);
    expiration_date = expiration_date . toGMTString ();

    // Expire the cookie
    var cookie_string = name + "=; expires=" + expiration_date;
    if (path != null)
        cookie_string += "; path=" + path;
    document . cookie = cookie_string;
}

function name_in_list (n, lst)
{
    var arr = lst . split ("; ");
    for (var j = 0; j < arr . length; j ++) {
        if (arr[j] == n)
            return true;
    }
    return false;
}

function delete_all_cookies (path, exceptions)
{
    // Get cookie list and split into an array of cookie entries
    var cookie_string = "" + document . cookie;
    var cookie_array = cookie_string . split ("; ");

    // Delete each cookie ...

    //   EXCEPT those whose naems appear in the semicolon delimited list
    //   passed in as the second parameter to this function
    for (var i = 0; i < cookie_array . length; ++ i) {
        var single_cookie = cookie_array [i] . split ("=");
        var name = single_cookie [0];
         if (name_in_list(name, exceptions) == false)
             delete_cookie (name, path);
     }
}

// -->
</script>

<html>
<head>
<meta http-equiv="Content-Type" content=
"text/html; charset=UTF-8">
<title>PKMS Administration: User Log Out</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" onLoad=delete_all_cookies("/",exception_list)>
<font size="+2"><b>User %USERNAME% has logged out.</b></font>
</body>
</html>
```

Modifying ConfigService.properties

To modify the WebSphere Portal logout command to point to the WebSEAL logout command, use these steps:

1. Navigate to the following directory on the Portal Server node:

   ```
   # /opt/WebSphere/PortalServer/shared/app/config/services
   ```

2. Back up the ConfigService.properties file to ConfigService.properties.org.
3. Modify the ConfigService.properties with the contents shown in Example 3-15. Update the redirect.logout and redirect.logout.ssl to true, and redirect.logout.url for your environment.

   > **Note:** Example 3-15 includes only the values that we changed, not the entire contents of the ConfigService.properties file.

Example 3-15 ITSO example ConfigService.properties snippet

```
# Logout redirect parameters
#
# Default: false, false, <none>
redirect.logout     = true
redirect.logout.ssl = true
redirect.logout.url = https://wslx1.itso.ral.ibm.com/pkmslogout?filename=wpslogout.html
```

4. Save and close the file.

Modifying ToolBarInclude.jsp files

This section modifies the ToolBarInclude.jsp file to address the following items:

- The ability of end users to self-register by removing the link to the self-registration window from the public portal page
- The ability of end users to edit their profiles by removing the link to the edit profiles window from the private portal page
- The ability of end users to request their passwords by removing the forgot password link from the public portal page
- The address of the Log in link by altering the link to point to the Reverse Proxy node

The procedure requires that each ToolBarInclude.jsp file is updated in each theme in which the file exists.

> **Note:** There is one occurrence of the ToolBarInclude.jsp in the root of the ../wps.war/themes/html directory. This ToolBarInclude.jsp is used as the default if a theme does not have its own ToolBarInclude.jsp.
>
> In addition, you need to update the ToolBarInclude.jsp (if found) in the ../wps.ear/wps.war/themes/html/<theme_name> directory for each theme.

Modify the ToolBarInclude.jsp for each occurrence of the file:

1. Navigate to the either of the following directories on the Portal Server node:

   – Root directory:

     /opt/WebSphere/AppServer/installedApps/wpslx1/wps.ear/wps.war/themes/html

   – Theme directory:

     /opt/WebSphere/AppServer/installedApps/wpslx1/wps.ear/wps.war/themes/html/*theme*

   Here, *theme* is the theme directory of interest. We listed the themes that we updated to the ToolBarInclude.jsp:

   – Admin
   – AdminLeftNavigation
   – Corporate
   – Engineering
   – Finance

   - Science
   - YourCoFinancial
   - YourCoFinancial2

2. Back up the ToolBarInclude.jsp.
3. Open the file in a text editor. Comment out the forgot password, selfcare, and enroll buttons as shown Example 3-16.
4. Edit the login button section as shown in Example 3-16 for each theme used.

Example 3-16 ITSO example ToolBarInclude.jsp snippet to be used as the default theme

```
<%-- forgot password button --%>
<%--
<wps:if loggedIn="no" notScreen="ForgotPassword">
            <td class="wpsToolBar" valign="middle" align="<%=bidiAlignRight%>" nowrap>
                <a class="wpsToolBarLink" href='<wps:url screen="ForgotPassword"
                home="public"/>'><wps:text key="link.password" bundle="nls.engine"/></a>
            </td>
</wps:if>
--%>

<%-- selfcare button --%>
<%--
<wps:if loggedIn="yes" notScreen="SelfcareUserForm,SelfcareUserConf" portletSolo="no">
            <td class="wpsToolBar" valign="middle" align="<%=bidiAlignRight%>" nowrap>
                <a class="wpsToolBarLink" href='<wps:url command="PrepareSelfcare"
                reqid="no"/>'><wps:text key="link.selfcare" bundle="nls.engine"/></a>
            </td>
</wps:if>
--%>

<%-- enroll button --%>
<%--
<wps:if loggedIn="no">
            <%
                String dt = com.ibm.wps.puma.UserManager.instance().getDirectoryType();
                if (dt==null)
                {
                    dt = "";
                }
                if (!dt.equals("SSPM"))
                {
            %>
              <td class="wpsToolBar" valign="middle" align="<%=bidiAlignRight%>" nowrap>
                <a class="wpsToolBarLink" href='<wps:url command="PrepareEnrollment"
home="public" reqid="no"/>'><wps:text key="link.enrollment" bundle="nls.engine"/></a>
              </td>
            <%
```

```
               }
            %>
</wps:if>
--%>

<%-- Edit the login button section as follows: --%>
<%-- login button --%>
<wps:if loggedIn="no" notScreen="Login">
               <td class="wpsToolBar" valign="middle" align="<%=bidiAlignRight%>" nowrap>
                <a class="wpsToolBarLink" href='<wps:url home="protected" screen="Home"
ssl="true">'><wps:text key="link.login" bundle="nls.engine"/></a>
               </td>
</wps:if>
```

5. Save and close the file.
6. Open and save the version of Default.jsp found in the root of the html directory and corresponding theme directory (if they exist):

   ```
   /opt/WebSphere/AppServer/installedApps/wpslx1/wps.ear/wps.war/themes/html/D
   efault.jsp
   ```

   This is necessary to make the application server recompile the JavaServer Pages (JSP) to have the changes take effect (touch operation).

Modifying WpsHostName property in wpconfig.properties

After junction configuration, the property WpsHostName in wpconfg.properties should be set to the WebSEAL host name. This allows proper URL filtering and protocol switching by WebSEAL.

1. Navigate to the following directory:

   ```
   # cd /opt/WebSphere/PortalServer/config
   ```

2. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file:

   ```
   # ./WPSconfig.sh backup-main-cfg-file
   ```

3. Modify the wpconfig.properties file as follows:

   From (Portal Server node host name):

   ```
   WpsHostName=wpslx1.itso.ral.ibm.com
   ```

   To (Reverse Proxy node host name):

   ```
   WpsHostName=wslx1.itso.ral.ibm.com
   ```

4. Save and close the file.
5. Execute the **WPSconfig.sh** command to load the configuration changes:

   ```
   # cd /opt/WebSphere/PortalServer/config
   # ./WPSconfig.sh httpserver-config
   ```

6. Restart the WebSphere_Portal application server:

```
# /opt/WebSphere/AppServer/bin/stopServer.sh WebSphere_Portal -username
wpsbind -password wpsbind

# /opt/WebSphere/AppServer/bin/startServer.sh WebSphere_Portal
```

Verifying the creation and login of a new user

To further test the TAI, create a new user (any random user) as follows:

1. Open a console window on the Reverse Proxy node.
2. Start a **pdadmin** session by logging on as follows:

   ```
   # pdadmin -a sec_master
   ```

   When prompted, enter the `sec_master` password.
3. Enter the following commands from the **pdadmin** console to create a user via Tivoli Access Manager. The syntax of the user create command is:

   ```
   user create -gsouser user_name user_dn common name surname password
   ```

   In our example, the command is:

   ```
   user create -gsouser bob uid=bob,cn=users,dc=itso,dc=ibm,dc=com “Bob”
   “Robert” “password0”

   user modify bob account-valid yes
   ```

   > **Tip:** The user password chosen must conform to the Tivoli Access Manager password policy. It requires a minimum length of eight characters formed by at least four alphabetical characters and one non-alphabetical character with no more than two repeated characters. The password policy can be changed on a user base or global base.
   >
   > For more information, refer to the *Base Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1360.

4. Log in to the portal using this newly created user:

   ```
   https://wslx1.itso.ral.ibm.com/portal/wps/myportal
   ```

3.7 Configuring Portal authorization with TAM

This section explains how to implement authorization using Tivoli Access Manager for WebSphere Portal. It includes an example for externalizing the WebSphere Portal YourCo Financial portlet page resources for authorization via Tivoli Access Manager.

This procedure requires the following tasks:

1. Configuring SSL between WebSphere and TAM
2. Implementing JAAS authentication
3. Modifying WebSphere Portal configuration files
4. Verifying entries in TAM for Portal external authorization
5. Externalizing a resource

3.7.1 Configuring SSL between WebSphere and TAM

The **SrvSslCfg** command helps to configure the SSL connection between the WebSphere Application Server and Tivoli Access Manager. This command creates a keyfile and a properties file, which is used later for the WebSphere Portal configuration. The **SrvSslCfg** command also creates the user who is specified as *was_id* and inserts this user in the following Tivoli Access Manager LDAP groups:

```
cn=remote-acl-users
cn=SecurityGroup,secauthority=Default
```

> **Note:** This step is required for application portlets that call the Tivoli Access Manager authorization API.

To configure the SSL connection between WebSphere Application Server used by WebSphere Portal and the Tivoli Access Manager, follow these steps:

1. Ensure that the following servers are started:
   - Directory Server node
     - Tivoli Directory Server
   - Policy Server node
     - Policy server
     - Authorization server
   - Reverse Proxy node
     - WebSEAL
   - Portal Server node
     - IBM HTTP Server
     - server1 application server
     - WebSphere_Portal application server
2. Open a console window as a root user on the Portal Server node.
3. In our example, create a script file named wpslx1_svrsslcfg.sh (see Example 3-17) in the /tmp/9121code/config/wps directory of the ITSO sample code containing the **SvrSslCfg** command and parameters for the ITSO

environment. In Example 3-17, update <password> with the correct sec_master password.

Example 3-17 ITSO sample wpslx1_svrsslcfg.sh

```
#!/bin/ksh

/opt/WebSphere/AppServer/java/jre/bin/java com.tivoli.pd.jcfg.SvrSslCfg -action
config -admin_id sec_master -admin_pwd <password> -appsvr_id amwas -port 7201
-mode remote -policysvr tamlx1.itso.ral.ibm.com:7135:1 -authzsvr
tamlx1.itso.ral.ibm.com:7136:1 -cfg_file
/opt/WebSphere/AppServer/java/jre/PdPerm.properties -key_file
/opt/WebSphere/AppServer/java/jre/lib/security/pdperm.ks -cfg_action create
```

In Example 3-17, note the following explanation:

– *admin_id* is the Tivoli administrator user ID. The default value is sec_master.

– *password* is the password for the Tivoli administrator user ID.

– *was_id* is the unique WebSphere Application Server identifier to be inserted into Tivoli Access Manager.

> **Note:** The value is user defined. In our example, we entered `amwas` for the *was_id*.

– *port_number* is the TCP/IP port that WebSphere Application Server listens to for policy server notifications. This value must be filled in even though WebSphere Application Server does not currently use this port.

– *pdmgrd_host* is the pdmgrd host (pdmgrd is a Tivoli Access Manager process).

– *pdacld_host* is the pdacld host (pdacld is a Tivoli Access Manager process).

– *pdacld_host* is the pdmgrd port. The default port number is 7135.

– *pdacld_port* is the pdacld port. The default port number is 7136.

– *rank* is the rank of the host. If only one host is specified, this value is 1.

– *config_file_path* is the path to the properties file that you create and insert into ExternalAccess ControlService.properties.

– *keystore_path* is the path to the keystore.

– replace is used to replace the existing configFile URL and keystoreFile URL.

Note: Alternatively, you can enter the following **SvrSslCfg** command to configure an SSL connection between the WebSphere Application Server on the Portal Server node and Tivoli Access Manager on the Policy Server node:

```
/opt/WebSphere/AppServer/java/jre/bin/java com.tivoli.pd.jcfg.SvrSslCfg
-action config -admin_id <admin_id> -admin_pwd <password> -appsvr_id
<was_id> -port <port_number> -mode remote -policysvr
<pdmgrd_host>:<pdmgrd_port>:<rank> -authzsvr
<pdacld_host>:<pdacld_port>:<rank> -cfg_file "<config_file_path>"
-key_file "<keystore_path>" -cfg_action replace
```

4. Execute the wpslx1_svrsslcfg.sh file.

   a. Copy the ITSO sample wpslx1_svsslcfg.sh to the /tmp directory.

   b. Change the privileges on the file:

   ```
   # chmod 777 /tmp/wpslx1_svrsslcfg.sh
   ```

   c. Execute the script file. Enter:

   ```
   # ./wpslx1_svrsslcfg.sh
   ```

   When the **SvrSslcfg** command completes, you should see the message `The configuration completed successfully`.

   Attention: We found problems running the **SvrSslCfg** command if Tivoli Access Manager V5.1 was configured before installing Tivoli Access Manager V5.1 Fixpack 4. To resolve this problem, unconfigure Tivoli Access Manager after the Fixpack 4 installation, and reconfigure Tivoli Access Manager.

5. To verify that the **SvrSslCfg** command worked properly, complete these tasks:

   a. Open a command window on the Policy Server node.

   b. Enter the following command to log in to **pdadmin**:

   ```
   pdadmin -a sec_master -p <password>
   ```

   c. Enter the following command to list the servers defined in Tivoli Access Manager:

   ```
   server list
   ```

   You should see the newly created server `amwas-wpslx1`, where *amwas* is the appsvr_id and *wpslx1* is the host name of the Portal Server node (WebSphere Application Server is installed).

3.7.2 Implementing JAAS authentication

This section explains how to configure the WebSphere Portal Server to extract and cache the Tivoli Access Manager JAAS credential from the HTTP header data sent by WebSEAL. This is required if you intend to call the JAAS API from within a portlet or if you intend to configure WebSphere Portal for external authorization using Tivoli Access Manager.

Modifying WebSphere Portal Server config files

Modify the WebSphere Portal Server configuration files to enable JAAS as follows:

1. Modify ConfigService.properties.
   a. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory on the Portal Server node.
   b. Modify the ConfigService.properties file as follows:
   ```
   execute.portal.jaas.login=true
   ```
   c. Save and close the file.
2. Modify callbackheaderslist.properties.‘
   a. Navigate to the /opt/WebSphere/PortalServer/shared/app/config directory on the Portal Server node.
   b. Modify the callbackheaderslist.properties file by uncommenting the following entries:
   ```
   header.1=iv-user
   header.2=iv-creds
   ```
   c. Save and close the file.
3. Modify ExternalAccessControlService.properties.
   a. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory on the Portal Server node.
   b. Modify the ExternalAccessControlService.properties file as follows:
   ```
   externalaccesscontrol.pdurl=file:/opt/WebSphere/AppServer/java/jre/PdPer
   m.properties
   ```
   c. Save and close the file.

Configuring JAAS in WebSphere Application Server

This section explains how to add a Tivoli Access Manager specific subclass of `java.security.Principal` to the WebSphere Application Server JAAS subject, which is used for the access control integration with Tivoli Access Manager.

JAAS can be configured within WebSphere Application Server on the Portal Server node by using the WebSphere Administrative Console or by using a WebSphere JACL command script.

> **Note:** As an alternative to this procedure, you can use the Administrative Console. We include a JACL command script with the ITSO example code to configure JAAS within the WebSphere Application Server for WebSphere Portal:
>
> ```
> /tmp/9121code/config/wps/config.was.jaas.jacl
> ```
>
> Ensure that server1 on the Portal Server node is running. Then enter the following command to execute the JACL command script:
>
> ```
> cd /opt/WebSphere/AppServer/bin
>
> wsadmin -f "/tmp/9121code/config/wps/config.was.jaas.jacl" -user wpsbind
> -password <password>
> ```

This section explains how to create four new JAAS Login Modules that add Tivoli Access Manager specific subclasses of java.security.Principal to the WebSphere Portal JAAS subject. To configure JAAS within the WebSphere Application Server on the Portal Server node using the WebSphere Administrative Console, follow these steps:

1. Ensure that WebSphere Application Server server1 is started.
2. Start the WebSphere Administrative Console and log in.
   a. Enter the following URL in a Web browser:

      `https://wplx1.itso.ral.ibm.com:9043/admin`

   b. Enter the WebSphere administrator credentials (for example, wpsbind).
3. Add `com.ibm.wps.sso.WebSealLoginModule` to the Portal_Login Application Login Configuration.
   a. From the WebSphere Application Server Administrative Console, select **Security →JAAS Configuration →Application Logins**.
   b. On the Application Login Configuration page, click **Portal_Login**.
   c. On the Portal Login page, in the Additional Properties section, click **JAAS Login Modules**.
   d. On the JAAS Login Modules page, click **New**.
   e. On the New JAAS Login Modules page, complete the following tasks:
      i. For Module Classname, type `com.ibm.ws.security.common .auth.module.proxy.WSLoginModuleProxy` (all on one line without any spaces or breaks).

   ii. For Authentication Strategy, select **REQUIRED**.
   iii. Click **OK**.

f. On the JAAS Login Modules page, under Module Classname, select **com.ibm.ws.security.common.auth.module.proxy. WSLoginModuleProxy**.

g. Scroll down the page. In the Additional Properties section, click **Custom Properties**.

h. Click **New**.

i. On the New Custom Properties page, complete the following tasks:

   i. For Name, type `delegate`.
   ii. For Value, type `com.ibm.wps.sso.WebSealLoginModule`.
   iii. Click **OK**.

4. Add `com.tivoli.mts.PDLoginModule` to the Portal_Login Application Login Configuration.

   a. From the WebSphere Application Server Administrative Console, select **Security** →**JAAS Configuration** →**Application Logins**.

   b. On the Application Login Configuration page, click **Portal_Login**.

   c. On the Portal Login page, in the Additional Properties section, click **JAAS Login Modules**.

   d. On the JAAS Login Modules page, click **New**.

   e. On the New JAAS Login Modules page, complete the following tasks:

      i. For Module Classname, type `com.ibm.ws.security.common.auth. module.proxy.WSLoginModuleProxy` (all on one line without any spaces or breaks).
      ii. For Authentication Strategy, select **REQUIRED**.
      iii. Click **OK**.

   f. On the JAAS Login Modules page, under Module Classname, select **com.ibm.ws.security.common.auth.module.proxy. WSLoginModuleProxy** (ensure that you click the second listed module).

   g. In the Additional Properties section, click **Custom Properties**.

   h. Click **New**.

   i. On the New Custom Properties page, complete these tasks:

      i. For Name, type `delegate`.
      ii. For Value, type `com.tivoli.mts.PDLoginModule`.
      iii. Click **OK**.

   If you now navigate to the Application Login Configuration →Portal_Login → JAAS Login Modules page, you see the two module classnames listed.

Note: Even though the JAAS Login Modules have the same module classname, they are still unique given their Name:Value pair in Custom Properties.

5. Add the `com.ibm.wps.sso.WebSealLoginModule` to the Portal_SubjectRebuild Application Login Configurations.
   a. From the WebSphere Application Server Administrative Console, select **Security →JAAS Configuration →Application Logins**.
   b. On the Application Login Configuration page, click **Portal_SubjectRebuild**.
   c. On the Portal_SubjectRebuild page, in the Additional Properties section, click **JAAS Login Modules**.
   d. On the JAAS Login Modules page, click **New**.
   e. On the New JAAS Login Modules page, complete the following tasks:
      i. For Module Classname, type `com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy` (all on one line without any spaces or breaks).
      ii. For Authentication Strategy, select **REQUIRED**.
      iii. Click **OK**.
   f. From the JAAS Login Modules page, under Module Classname, select **com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy**.
   g. In the Additional Properties section, click **Custom Properties**.
   h. Click **New**.
   i. On the New Custom Properties page, complete these tasks:
      i. For Name, type `delegate`.
      ii. For Value, type `com.ibm.wps.sso.WebSealLoginModule`.
      iii. Click **OK**.
6. Add the com.tivoli.mts.PDLoginModule to the Portal_SubjectRebuild Application Login Configuration.
   a. From the WebSphere Application Server Administrative Console, select **Security →JAAS Configuration →Application Logins**.
   b. On the Application Login Configuration page, click **Portal_SubjectRebuild**.
   c. On the Portal_SubjectRebuild page, in the Additional Properties section, click **JAAS Login Modules**.
   d. On the JAAS Login Modules page, click **New**.

e. On the New JAAS Login Modules page, complete the following tasks:

   i. For Module Classname, type `com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy` (all on one line without any spaces or breaks).
   ii. For Authentication Strategy, select **REQUIRED**.
   iii. Click **OK**.

f. On the JAAS Login Modules page, under Module Classname, click **com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy** (ensure that you click the second listed module).

g. In the Additional Properties section, click **Custom Properties**.

h. Click **New**.

i. On the New Custom Properties page, complete these tasks:

   i. For Name, type `delegate`.
   ii. For Value, type `com.tivoli.mts.PDLoginModule`.
   iii. Click **OK**.

If you now navigate to the Application Login Configuration → Portal_SubjectRebuild →JAAS Login Modules page, you see the two module classnames listed.

> **Note:** Even though the JAAS Login Modules have the same module classname, they are still unique given their Name:Value pair in Custom Properties.

7. Click **Save**.
8. On the Save to Master Configuration page, click **Save**.
9. Click **Logout**.

   Normally, we now restart the WebSphere_Portal application server. However, we defer this task until after we modify the WebSphere Portal configuration files.

3.7.3 Modifying WebSphere Portal configuration files

This section includes the following modifications to WebSphere Portal configuration files to enable external authorization using Tivoli Access Manager:

- Modifying the ExternalAccessControlService.properties
- Modifying AccessControlConfigService.properties
- Modifying services.properties
- Modifying AccessControlDataManagementService.properties
- Reordering the role names (optional)

Modifying the ExternalAccessControlService.properties

Modify the ExternalAccessControlService.properties file as follows:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory.
2. Open the ExternalAccessControlService.properties file.
3. Example 3-18 shows the uncommented directives and ITSO example values in the ExternalAccessControlService.properties file. You may need to update this file accordingly with values for your environment.

Example 3-18 ITSO example ExternalAccessControlService.properties

```
# ------------------------------------------------- #
# Properties of the External Access Control Service #
# ------------------------------------------------- #

## This flag indicates whether the configuration in this file
## has been configured to connect to the External Security Manager
externalaccesscontrol.ready=true

## Rolenames representations are qualified with a context built
## by the following parameters. For example, the Administrator@External_Access_Control/xxx/xxx
## is represented in the following ways:
##
## TAM: Protected object space entry
##    /WPSv5/Administrator@External_Access_Control/xxx/xxx/WPS/WebSphere_Portal/cell
##
## SiteMinder:
##    resource/subrealms under Domain: WebSphere Portal v5
##    /cell/WebSphere_Portal/WPS/Administrator@External_Access_Control/xxx/xxx
externalaccesscontrol.server=WebSphere_Portal
externalaccesscontrol.application=WPS
externalaccesscontrol.cell=cell


## ---------------------------------------
## Access Manager configuration
## ----------------------------------------

## After completing the PDJRTE and SrvSslCfg configuration,
## the following directives are needed to
## allow WP to use Access Manager as an External Security Manager

## Provide the root of your Protected Object Space for Portal Server entries
externalaccesscontrol.pdroot=/WPSv5

## Provide and administrative user and password with adequate rights in
## Tivoli to create, delete, modify the objects in the Protected Object Space.
## You can use the WAS PropFilePasswordEncoder utility to mask the password.
## Using PropFilePasswordEncoder will remove any comments and uncommented properties,
## so create a backup copy of this file for future reference.
## Example: <WAS_ROOT>/bin/PropFilePasswordEncoder
## <WPS_ROOT>/shared/app/config/services/ExternalAccessControlService.properties
## externalaccesscontrol.pdpw
## *NOTE* this command is on 3 lines in this file, but should be typed on 1 line
## in a command window.
externalaccesscontrol.pduser=sec_master
externalaccesscontrol.pdpw=<password>

## Specify the location of the Access Manager propeties file for PDJRTE
## This URL must be in the format file:///<path to properties file. http://
## urls are not supported.
externalaccesscontrol.pdurl=file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties

## (optional) Specify whether to create ACLs in  Access Manager for roles stored externally
## If this value is set to false, the Access Manager administrator will be responsible
## for all ACL linkages between TAM and WP
## values:
## true - if an TAM ACL will be created for EVERY resource
##    false - if no ACLs will be created for WP objects
externalaccesscontrol.createAcl=true

## (optional) Specify the action group and the customized actions to map to Portal
## role membership. If these items do not exist, they will be created at startup
## default values:
## externalaccesscontrol.pdactiongroup=[WPS]
## externalaccesscontrol.pdAction=m
externalaccesscontrol.pdactiongroup=[WPS]
externalaccesscontrol.pdaction=m
```

4. Save and close the file.

5. Use the WebSphere Application Server encoding mechanism to mask the password that will appear in the ExternalAccessControlService.properties file, by performing the following steps:

   a. Use the WebSphere Application Server encoding mechanism to mask passwords in the live version of the ExternalAccessControlService.properties file.

      The following command masks the sensitive fields and removes all comments from the file. The original version of the file with the password in the clear and all comments intact is preserved with a bak extension.

      `/opt/WebSphere/AppServer/bin/PropFilePasswordEncoder.bat` *`filename property_name`*

Consider this example:

```
/opt/WebSphere/AppServer/bin/PropFilePasswordEncoder.bat
/opt/WebSphere/PortalServer/shared/app/config/services/ExternalAccessCon
trolService.properties externalaccesscontrol.pdpw
```

The command generates the *filename*.bak, for example, ExternalAccessControlService.properties.bak.

b. The next time you need to change the password, follow these steps:

   i. Copy the backup version of the file over the live version.

   ii. Edit this new live file as needed and enter the new password in clear text.

   iii. Save the file.

   iv. Run the WebSphere Application Server mechanism on the file.

c. For security reasons, either remove the password from the backup file with the clear text password (for example, ExternalAccessControlService.properties.bak), or delete the file.

Modifying AccessControlConfigService.properties

To modify the AccessControlConfigService.properties file, follow these steps:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory.
2. Make a backup copy of the AccessControlConfigService.properties.
3. Modify the following value in the AccessControlConfigService.properties file:

   ```
   accessControlConfig.enableExternalization=true
   ```

4. Save and close the file.

Modifying services.properties

To modify the services.properties file, perform the following steps:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config directory.
2. Make a backup copy of the services.properties.
3. Modify the services.properties file as shown in Example 3-19. You need to modify only the last entry in Example 3-19. Note of the value marked in bold.

> **Note:** Some values in Example 3-19 wrap to the following line.

Example 3-19 ITSO example services.properties snippet

```
com.ibm.wps.ac.impl.AccessControlDataManagementService=com.ibm.wps.ac.impl.AccessControlDataMan
agementServiceImpl
com.ibm.wps.services.ac.PermissionFactoryService=com.ibm.wps.ac.impl.PermissionFactoryImpl
com.ibm.wps.services.ac.ACPrincipalFactoryService=com.ibm.wps.ac.impl.ACPrincipalFactoryImpl
com.ibm.wps.services.ac.internal.AccessControlConfigService=com.ibm.wps.ac.impl.AccessControlCo
nfigImpl
com.ibm.wps.services.ac.AccessControlService=com.ibm.wps.ac.impl.AccessControlImpl
com.ibm.wps.services.ac.ExternalAccessControlService=com.ibm.wps.ac.esm.TAMExternalAccessContro
lImpl
```

4. Save and close the file.

Modifying AccessControlDataManagementService.properties

To modify the AccessControlDataManagementService.properties file, follow these steps:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory.
2. Make a backup copy of the AccessControlDataManagementService.properties file.
3. Modify the following values in the AccessControlDataManagementService.properties file:

   ```
   accessControlDataManagement.enableNestedGroups=false
   accessControlDataManagement.cacheTimeout=30
   ```

4. Add the following entries at the bottom of the file if they do not exist:

   ```
   accessControlDataManagement.externalizeAllRoles=false
   accessControlDataManagement.createAdminMappingXMLAccess=true
   ```

   Note the following explanation of the parameters:

   – **cacheTimeout**: Portal Access Control maintains caches for better performance of requests. When a role mapping is changed in the external system, Portal Access Control does not know about these changes unless the affected user or users log out and in again. This property automatically invalidates the Portal Access Control caches after a given time (in seconds).

   – **externalizeAllRoles**: This property is applicable only for externalization of resources through the user interface. If the property is set to *false* and a resource is externalized, the following happens:

     i. The resource and all descendants of this resource that are not private and not externalized so far are externalized.

ii. The roles (and the role mappings) that exist on all resources that were identified in the previous step are written into the external security manager object space.

iii. For the root resource that was chosen to be externalized, a role mapping for the Administrator role for the executing user is created in the external security manager object space.

If this property is set to *true*, then in addition to the previous three steps, roles are created in the external security manager object space for all action sets for the root resource that have not already been created in steps ii and iii.

- **createAdminMappingXMLAccess**: This property is applicable only for externalization of resources through XMLAccess. If the property is set to *false* and a resource is externalized, the following happens:

  i. The resource is externalized.

  ii. The roles that exist (and the role mappings) on the resource are written into the external security manager object space.

  If the property is set to *true*, then in addition to the previous two steps, a role mapping for the Administrator role is created for the executing user in the external security manager object space.

- **enableNestedGroups**: Tivoli Access Manager V5.1.0.2 does not support nested groups. WebSphere Portal V5.0.2 does support nested groups, but due to the Tivoli Access Manager limitation, we must set this to *false* (do not use nested groups).

5. Save and close the file.
6. Restart the WebSphere_Portal application server.

> **Note:** You can defer this step if you are planning to implement the reorderRoleNames property.

When WebSphere Portal starts, TAMExternalAccessControlServices creates the necessary topology in Tivoli Access Manager to begin externalizing roles. It also creates the Administrator@EXTERNAL ACCESS CONTROL/1 role. Depending on your configuration setting for externalaccesscontrol.pd_createAcl, it adds the *wpsadmin* user to the ACL that is attached to this role.

Reordering the role names (optional)

By default, externalized roles appear in the external security manager as Role Type@Resource Type/Name/Object ID, for example:

```
Administrator@PORTLET_APPLICATION/Welcome/1_1_1G
```

You can change this format to Resource Type/Name/Object ID@Role type. This format change groups the roles by resource name instead of by role type, for example:

```
PORTLET_APPLICATION/Welcome/1_0_1G@Administrator
```

This format change is visible only when the roles are externalized. This change does not affect the way roles are displayed in WebSphere Portal. The Administrator@VIRTUAL/wps.EXTERNAL ACCESS CONTROL/1 role is never affected by this format change. This role always appears with the role type "Administrator" on the left.

To reorder the role names when listed, perform the following steps on the Portal Server node:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory.
2. Make a backup copy of the AccessControlDataManagementService.properties file.
3. Modify the AccessControlDataManagementService.properties file as follows:

   ```
   accessControlDataManagement.reorderRoleNames=true
   ```

   If the property does not exist, add it.
4. Save and close the file.
5. Restart the WebSphere_Portal application server:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./stopServer.sh WebSphere_Portal -user wpsbind -password wpsbind
   # ./startServer.sh WebSphere_Portal
   ```

3.7.4 Verifying entries in TAM for Portal external authorization

When the WebSphere Portal starts, TAMExternalAccessControlServices creates the necessary topology in Tivoli Access Manager to begin externalizing roles. It also creates the core WPS_Administrator-Virtual_wps-EXTERNAL ACCESS CONTROL_1 role. And it creates an access control list (ACL). And it adds the *wpsadmin* user to the ACL. It attaches this ACL to the core role.

To confirm this, execute the following commands in a **pdadmin** session on the Reverse Proxy node:

1. Open a command window and log in to **pdadmin**:

   ```
   pdadmin -a sec_master -p <password>
   ```

2. Verify that the /WPSv5 object space has been created by entering the following command:

   ```
   objectspace list
   ```

   You should see the /WPSv5 object space in the list.

3. Verify that the WPS action group is created by entering the following command:

   ```
   action group list
   ```

   You should see the WPS action group in the list.

4. To verify that the portal administrator has the Administrator role, view the ACL for the namespace entry that represents the Administrator@VIRTUAL/EXTERNAL ACCESS CONTROL_1 role by entering the following command on the **pdadmin** command line:

   ```
   acl show WPSv5_Administrator-VIRTUAL_wps-EXTERNAL_ACCESS_CONTROL_1
   ```

   You should see the following new entry:

   ```
   User wpsadmin [WPS]m
   ```

   > **Note:** Alternatively, use the Tivoli Access Manager Web Portal Manager to view the ACLs.

3.7.5 Externalizing a resource

As part of Chapter 4, “Deploying the secure portal application” on page 165, we demonstrate how to externalize a resource from WebSphere Portal to be managed by Tivoli Access Manager for authorization.

3.8 Additional configuration

This section includes addition configuration tasks that may be optional for your business requirements.

3.8.1 Configuring WebSEAL and WebSphere Portal session timeouts

Both WebSEAL and WebSphere Portal establish sessions with the client browser when initially accessed. These sessions are independent of each other, but both must be valid in order for the system to work as expected.

For a detailed description of the architecture and considerations for session timeouts refer to the “Design and integration guidelines” chapter in *Develop and*

Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1, SG24-6325.

This section explains the following tasks to configure WebSphere Portal and WebSEAL session timeouts:

1. Modifying the WebSphere Portal session timeout
2. Configuring WebSphere Portal to resume timed out sessions
3. Modifying the WebSEAL session timeout

Modifying the WebSphere Portal session timeout

Modify the WebSphere Portal session timeout values on the Portal Server node from the WebSphere Application Server Administrative Console, as follows:

1. Ensure that the server1 application server is started. If not, start the server.

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh server1
   ```

2. Start the WebSphere Application Server Administration Console by entering the following URL in a Web browser:

   `http://was_hostname:9090/admin`

3. Log on to the WebSphere Administration Console (for example, wpsbind).
4. Click **Servers** →**Application Servers**.
5. Click **WebSphere_Portal**.
6. Click **Web Container**.
7. Click **Session Management**.
8. From the Session Management Configuration page, complete the following tasks:

   a. Select **Set timeout**.
   b. For Minutes, type `25` (default 30).

   > **Note:** The Session timeout value (minutes) should be set based on the business requirements for security and performance. For example, consider reducing this value for high volume Web sites.

   c. Click **OK**.

9. Click **Save**.
10. On the Save to Master Configuration page, click **Save**.
11. Log out and close the WebSphere Administrative Console.
12. Restart the WebSphere_Portal application server for this change to take effect. However, if you plan to perform the steps in “Configuring WebSphere

Portal to resume timed out sessions”, you can defer the restart until after that task is complete.

Configuring WebSphere Portal to resume timed out sessions

To configure the WebSphere Portal to resume timed out sessions, complete the following steps on the Portal Server node:

1. Navigate to the /opt/WebSphere/PortalServer/shared/app/config/services directory.
2. Modify the ConfigService.properties file as follows:
   a. Search the ConfigService.properties file for `persistent.session.level`.

   > **Note:** We chose to accept the default `persistent.session.level = 0`, which means that the window state is not preserved when the session is persisted.

   b. Add the following entry above the persistent.session.level property:

   ```
   timeout.resume.session = true
   ```

   > **Note:** By default, WebSphere Portal invalidates sessions when they time out. The `timeout.resume.session = true` WebSphere Portal persists the sessions when they time out instead of invalidating them.
   >
   > For more information, refer to the “Design and integration guidelines” chapter in *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.

   c. Save and close the ConfigService.properties file.
3. Restart the WebSphere_Portal application server for this change to take effect.

Modifying the WebSEAL session timeout

Modify the WebSEAL session timeout values on the Reverse Proxy node, as follows:

1. Navigate to the /opt/pdweb/etc directory.
2. Modify the webseald-default.conf file.
   a. Search for `inactive-timeout`.
   b. We recommend that you change the inactive-timeout value from the default of 600 seconds to match the WebSphere Portal session timeout.

For example, we set the WebSphere Portal session timeout to 25 minutes. Therefore, the inactive-timeout for WebSEAL is set to 1500 seconds.

```
inactive-timeout = 1500
```

> **Note:** The inactive-timeout value should not exceed the WebSEAL timeout value (located above the inactive-timeout value in the .conf file). By default, timeout = 3600 seconds. Simply increase the timeout value to be equal to or greater than the inactive-timeout value.
>
> For more information, consult the following resources:
>
> - “Design and integration guidelines” chapter in *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325
> - *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359

3. Save and close the webseald-default.conf file.
4. Restart WebSEAL for this change to take effect.

3.8.2 Configuring WebSEAL to handle favicon.ico

Some Web browsers have unexpected behavior with the default WebSEAL configuration attempting to load favicon.ico.

To configure WebSEAL to handle favicon.ico to work with all Web browsers, perform the following steps on the Reverse Proxy node:

1. Open a command window and log into **pdadmin**.

   ```
   pdadmin -a sec_master -p <password>
   ```

2. Enter the following **acl attach** command:

   ```
   acl attach /WebSEAL/wslx1-default/favicon.ico WP_all_access
   ```

   Here *wslx1-default* is the WebSEAL instance and *WP_all_access* is the ACL that allows access for all users (defined in Example 3-11 on page 132).

4

Deploying the secure portal application

This chapter explains how to deploy the ITSO Bank secure portal application in the production environment. The ITSO Bank application is made up of a back-end application and a front-end portal application. The back-end application consists of Enterprise JavaBeans (EJB) complying with EJB 2.0 and Java 2 Platform, Enterprise Edition (J2EE) 1.3 specifications. It also packages a dynamic link library (DDL) file to create tables in the required application database. The portal application consists of portlets that are ready for deployment on WebSphere Portal Server V5.0.2.2.

4.1 ITSO Bank application overview

The ITSO Bank application is based on a secure eBanking portal scenario that provides personalized information and services to the customer, coupled with an administration interface for the bank officials.

This application provides the following personalized services to the user that belongs to the role of customer:

- View balance in the savings and checking accounts
- Transfer funds between savings and checking accounts

This application provides the following administrative services to the user that belongs to the role of manager:

- Enter customer information
- Modify customer information
- View balance in the savings and checking accounts of any customer
- View transaction history of customer accounts

4.2 Deploying the ITSO Bank back-end application

This section explains how to deploy the back-end application. It includes Click such activities as creating and setting up the ITSO Bank database. It also explains how to configure WebSphere Application Server so that it can host the ITSO Bank application.

The ITSO Bank back-end is a J2EE application that uses a DB2 database (or Cloudscape database in the development environment). The ITSO Bank back-end and database can be installed on a remote WebSphere Application Server. In our example, we deploy both the back-end J2EE application and front-end portlets on the Portal Server node. For the purposes of showing the proper configuration, we create a separate application server (for example, ITSOBankServer) to deploy the back-end application.

The deployment of the back-end application includes the following tasks:

1. Creating an application server
2. Downloading and unpacking the ITSO Bank sample code
3. Creating the ITSO Bank application database
4. Adding the ITSOid attribute to the LDAP schema
5. Creating the groups and users for the ITSO Bank application
6. Creating the ITSOBankDataSource data source
7. Deploying the back-end application EAR file

4.2.1 Creating an application server

This section explains how to create the ITSOBankServer application server used to deploy and run the ITSO Bank back-end enterprise application.

Creating a new application server

To create a new application server (for example, ITSOBankServer), perform the following steps on the Portal Server node:

1. Ensure that the server1 application server is started. server1 is the server where the WebSphere Application Server Administrative Console Enterprise application is installed. If not, start server1 from a command window as follows:

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh server1
   ```

2. Start the WebSphere Administrative Console by entering the following URL in a Web browser:

   ```
   http://wpslx1.itso.ral.ibm.com:9090/admin
   ```

3. Log on as the user ID `wpsbind`.
4. Select **Servers** →**Application Servers**.
5. Click **New**.
6. On the Create New Application Server page, complete the following tasks:

   a. For Select node, select **wpslx1/wpslx1**.
   b. For Server name, type `ITSOBankServer`.
   c. Accept the default settings for the remaining options.
   d. Click **Next**.

7. On the Confirm new application server page, review the settings and click **Finish**.
8. Click **Save**.
9. On the Save to Master Configuration page, click **Save**.

Determining an application server bootstrap port

When a new application server is created, a bootstrap port is generated that is unique based on existing applications servers. When the itsobank.properties file is modified in a later procedure, a boot strap port value is needed for the application server to deploy the ITSO Bank application for JNDI lookup.

To determine the ITSOBankServer application server boot strap port, follow these steps:

1. From the WebSphere Application Server Administrative Console, select **Servers →Application Servers**.
2. Click **ITSOBankServer**.
3. Scroll down the page, and click **End Points**.
4. Click **BOOTSTRAP_ADDRESS**.
5. On the BOOTSTRAP_ADDRESS page, record the value of the port (for example, 2811). Click **Cancel**.
6. Click **Log out**.

> **Note:** In our example, we used the port value that was generated by WebSphere Application Server for the newly created ITSOBankServer application server. Alternatively, you can change the port value to a new value.

4.2.2 Downloading and unpacking the ITSO Bank sample code

This section explains how to download the ITSO sample code 9121code.zip, unpack the ZIP file, and extract files from the Enterprise Archive (EAR).

1. Download the ITSO sample code 9121code.zip file at:

   ftp://www.redbooks.ibm.com/redbooks/REDP9121

   > **Note:** For a description of the ITSO sample code and where to download it, refer to Appendix C, “Additional material” on page 221.

2. Unpack the 9121code.zip file. For example, we unpacked the ZIP file to the /tmp/9121code directory.
3. Change privileges on the files:

   ```
   # chmod -R 777 /tmp/9121code
   ```

4. Create the directory /tmp/ITSOBankApp.
5. Copy the ITSOBankEAR.ear file from the ITSO sample code /tmp/9121code/deploy directory to the /tmp/ITSOBankApp directory.
6. Extract the ITSOBankEJB.jar file from the ITSOBankEAR.ear file.

   a. Open a command window and change to the /tmp/ITSOBankApp directory.

b. Extract the EJB JAR file:

```
/opt/WebSphere/AppServer/java/bin/jar -xvf ITSOBankEAR.ear
ITSOBankEJB.jar
```

7. Extract the Table.ddl file from the ITSOBankEJB.jar file.

   a. Open a command window and change to the /tmp/ITSOBankApp directory.

   b. Enter the following command to extract the files:

   ```
   /opt/WebSphere/AppServer/java/bin/jar -xvf ITSOBankEJB.jar
   META-INF/Table.ddl
   ```

   c. Remove the following comment from the Table.ddl file and save the file:

   ```
   -- Generated by Relational Schema Center on Fri May 14 17:58:18 EDT 2004
   ```

   > **Attention:** The previous comment is added while generating the Table.ddl from WebSphere Studio. We found that, when we attempted to create the tables using the **`db2 -tvf Table.ddl`** command on Linux, the command failed (this works OK on Windows). To work around this issue, we simply removed the comment line.

8. Change the privileges on the files:

   ```
   # chmod -R 777 /tmp/ITSOBankApp
   ```

4.2.3 Creating the ITSO Bank application database

Although you can perform the following steps through the DB2 Control Center and the DB2 Command Center, we chose to use the DB2 Command Line Processor. In our example, the DB2 Client is installed on the Portal Server node and the DB2 server is installed on the Directory Server node.

Perform the following steps on the Portal Server node (DB2 Client) to create the ITSO Bank database:

1. Start the DB2 Command window by logging on as the DB2 instance owner on the Directory Server node:

   ```
   # su - db2inst1
   ```

2. Attach to the DB2 server on the Directory Server node:

   ```
   db2 attach to ldaplx1 user db2inst1 using <password>
   ```

   > **Note:** We cataloged the tcpip node already in 2.5.10, “Configuring the DB2 UDB client/server node” on page 80.

3. Create the database ITSOBank. Enter:

```
db2 create db ITSOBank
```

4. Connect to the database:

```
db2 connect to ITSOBank user db2inst1 using <password>
```

5. Create the tables required for the application:

```
db2 -tvf /tmp/ITSOBankApp/META-INF/Table.ddl
```

4.2.4 Adding the ITSOid attribute to the LDAP schema

This section explains how to add the ITSOid attribute to the LDAP schema. The ITSOid is a correlation ID between the ITSO Bank portlet application and the user entries in the LDAP repository.

To add the ITSOid attribute to the LDAP schema, follow these steps:

1. Ensure that the server1 application server is started on the Policy Server node (location of Tivoli Directory Server Web Administration Tool).
2. Access the Tivoli Web Administration Tool from a Web browser:

   ```
   http://tamlx1.itso.ral.ibm.com:9080/IDSWebApp/IDSjsp/Login.jsp
   ```

3. On the Login to the Web Administration Tool page, complete these tasks:
   a. For LDAP host name, select **ldaplx1.itso.ral.ibm.com**.
   b. For Username, type `cn=root`.
   c. Type your password.
   d. Click **Login**.
4. Click **Schema Management** →**Add an attribute**.
5. On the Add an attribute page, complete the following tasks:
   a. For Attribute name, type `ITSOid`.
   b. For Description, type `Correlation ID between the ITSO Bank portlet application and the user entries in the LDAP repository`.
   c. For OID, type `ITSOid-OID`.
   d. At the bottom of the page, click **OK**.
6. Click **Schema Management** →**Manage object classes**.
7. On the Manage object classes page, complete these tasks:
   a. Scroll down to the bottom of the page and, from the drop-down list, select **Page 15 of 18 nsLiProfile**. Click **Go**.
   b. Select the **organizationalPerson** object class and then click **Edit** (found by scrolling to the right side of the page).
   c. On the Edit object class: organizationalPerson page, click **Attributes**.

   d. On the Attributes page, from the Available attributes drop-down list, select **ITSOid**. Click **Add to optional**.

   e. Scroll down the Attributes page and click **OK**.

8. Click **Logout** to log out of the Web Administration Tool.
9. Restart the IBM Tivoli Directory Server V5.2.

4.2.5 Creating the groups and users for the ITSO Bank application

This section explains how to create groups and users needed for the ITSO Bank application. We create a ITSOManagers and ITSOCustomers group. In addition, we create a manager user ID in the managers group. The manager user ID is required. However, the customer user IDs can be created from the ITSO Bank application.

To create the ITSOManagers and ITSOCustomers groups and manager1 user ID for the ITSO Bank application, perform the following steps:

1. Create an LDIF file like the one shown in Example 4-1 on the Policy Server node (location of Tivoli Directory Server client).

   For example, we created the itsobank.ldif file that will be imported to create the groups and users for the ITSO Bank. The itsobank.ldif file can be found in the ITSO sample code /tmp/9121code/deploy directory.

   If you copy this file from the /tmp/9121code/deploy directory, change file permissions as follows:

   ```
   # chmod 777 itsobank.ldif
   ```

   Change the userpassword value as desired for your environment.

Example 4-1 ITSO Bank application - itsobank.ldif

```
version: 1
# ITSO example: itsobank.ldif file

dn: uid=manager1,cn=users,dc=itso,dc=ibm,dc=com
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: manager1
userpassword: passw0rd
sn: Manager
givenName: ITSOBank
cn: ITSOBank Manager

dn: cn=ITSOManagers,cn=groups,dc=itso,dc=ibm,dc=com
```

```
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=manager1,cn=users,dc=itso,dc=ibm,dc=com
cn: ITSOManagers

# This is a dummy user entry for ITSOCustomers group.
dn: uid=ITSOBankDummyUser,cn=users,dc=itso,dc=ibm,dc=com
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: ITSOBankDummyUser
sn: Dummy
givenName: ITSOBank
cn: ITSOBank Dummy User

dn: cn=ITSOCustomers,cn=groups,dc=itso,dc=ibm,dc=com
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=ITSOBankDummyUser,cn=users,dc=itso,dc=ibm,dc=com
cn: ITSOCustomers
```

> **Note:** You must set a uniquemember attribute for the ITSOCustomers group at the time of its creation. We created the ITSOBankDummyUser ID. Although this user belongs to the ITSOCustomers group, it is not used at any point for the ITSOBank application.

2. Open a console window and enter the following command to import the groups and user into the Tivoli Directory Server:

   ```
   ldapadd -D cn=root -w <password> -h ldaplx1 -c -i itsobank.ldif
   ```

3. Import the ITSOManagers and ITSOCustomers groups and manager1 user ID into Tivoli Access Manager from a command window:

   a. Copy the ITSO sample code /tmp/9121code/deploy/itsobank.pd to the /tmp directory on the Policy Server node. Example 4-2 lists the contents of the itsobank.pd command file.

Example 4-2 ITSO Bank application itsobank.pd

```
group import ITSOManagers cn=ITSOManagers,cn=groups,dc=itso,dc=ibm,dc=com

group import ITSOCustomers cn=ITSOCustomers,cn=groups,dc=itso,dc=ibm,dc=com

user import manager1 uid=manager1,cn=users,dc=itso,dc=ibm,dc=com
user modify manager1 account-valid yes
```

b. Change the file permissions as follows:

```
# chmod 777 itsobank.pd
```

c. Open a console window and enter the following command to execute the import into Tivoli Access Manager:

```
pdadmin -a sec_master -p <password> itsobank.pd
```

4. Verify that you can log on to the WebSphere Portal via WebSEAL.

a. Enter the following URL in a Web browser:

```
https://wslx1.itso.ral.ibm.com/portal/wps/myportal
```

b. Log on as the `manager1` user ID and password (found in itsobank.ldif).

4.2.6 Creating the ITSOBankDataSource data source

To create the ITSOBankDataSource (used by the back-end application to access the ITSOBank database), perofrm the following steps on the Portal Server node:

1. Ensure that the ITSOBankServer application server is started.

   If it is not started, enter the following statement in a command window:

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh ITSOBankServer
   ```

2. Ensure that the server1 application server is started.

   server1 is the server where the WebSphere Application Server Administrative Console Enterprise application is installed. If it is not started, start server1 from a command window:

   ```
   # /opt/WebSphere/AppServer/bin/startServer.sh server1
   ```

3. Start the WebSphere Administrative Console. Enter the following URL in a Web browser:

   ```
   http://wpslx1.itso.ral.ibm.com:9090/admin
   ```

4. Log on as the user ID `wpsbind`.

5. Create an authentication alias used by the data source.

   The ITSO Bank application uses an authentication alias to connect to the database through the data source configured for a DB2-based Java Database Connectivity (JDBC) Provider. To configure this authentication alias in the WebSphere Administrative Console, use the following steps:

   a. Click **Security** →**JAAS Configuration** →**J2C Authentication Data**.

   b. On the J2C Authentication Data page, click **New** to create a new alias.

   c. On the New alias page appears, complete these tasks:

      i. For Alias, type `ITSOBankAlias`.
      ii. For User ID, type `db2inst1`. This is the DB2 instance owner used to create a database.
      iii. For Password, enter your DB2 password (*db2_password*).
      iv. For Description, type `ITSO Bank Alias`.
      v. Click **OK**.

6. Create a JDBC provider.

   a. Click **Resources →JDBC Providers**.

   b. On the JDBC Providers page, complete these tasks:

      i. For Node, type `wpslx1`.
      ii. For Server, type `ITSOBankServer`.
      iii. Click **Apply**.

   c. On the JDBC provider page, click **New**.

   d. From the JDBC Providers drop-down list, select **DB2 Legacy CLI-based Type 2 JDBC Driver**, and click **OK**.

   > **Note:** The default DB2 JDBC Provider is deprecated. For this reason, we chose to use the DB2 Legacy CLI-based Type 2 JDBC Driver.

   e. On the DB2 Legacy CLI-base Type 2 JDBC Drive page, complete these tasks:

      i. For Name, type `DB2 Legacy CLI-based Type 2 JDBC Driver for ITSO Bank Application`.
      ii. For Description, type `DB2 JDBC2-compliant Provider`.
      iii. Click **OK**.

7. Create the data source.

   a. From the JDBC Providers page, click the newly created JDBC provider **DB2 Legacy CLI-based Type 2 JDBC Driver for ITSO Bank Application**.

   b. Scroll down the page to Additional Properties and click **Data Sources**.

   c. On the Data Sources page, click **New**.

   d. On the New Data Sources page, complete the following tasks:

      i. For Name, type `ITSOBankDataSource`.
      ii. For JNDI Name, type `jdbc/ITSO`.
      iii. For Container managed persistence, select **Use this Data Source in container managed persistence (CMP)**.
      iv. For Description, type `DB2 JDBC DataSource for ITSOBank`.

   v. For Component-managed Authentication Alias, select **wpslx1/ITSOBankAlias**.
   vi. Accept the default values for the remaining options (as we did in this example).
   vii. Click **OK**.

   e. Click **ITSOBankDataSource**.
   f. Under Additional Properties, click **Custom Properties**.
   g. Click **databaseName** and set the value to `ITSOBANK`. Click **OK**.
   h. Click **Environment** →**Manage WebSphere Variables**.
   i. Ensure that the variable DB2_JDBC_DRIVER_PATH is set to the location of db2java.zip file, for example:

```
DB2_JDBC_DRIVER_PATH /opt/IBM/db2/V81/java
```

   j. Click **Save**.
   k. On the Save to Master Configuration page, click **Save**.

8. Test the connection to the database:
   a. Select **Resources** →**JBDC Providers**.
   b. Click **DB2 Legacy CLI-based Type 2 JDBC Driver for ITSO Bank Application**.
   c. Under Additional Properties, click **Data Sources**.
   d. Select **ITSOBankDataSource**, and then click **Test Connection**.

   You should see the following message if you are successful:

```
Test connection for the datasource ITSOBankDataSource on server server1
at node wpslx1 was successful.
```

Note: We found that we had to restart the application server before the datasource test connection worked properly.

4.2.7 Deploying the back-end application EAR file

To deploy the ITSOBankEAR.ear file, perform the following steps:

1. Click **Applications** →**Install New Application**.
2. On the Prepare for the application installation page, complete these tasks:
   a. Select **Local path**.
   b. For Local path, type `/tmp/ITSOBankApp/ITSOBankEAR.ear`.
   c. Click **Next**.
3. Click **Next**.

4. We accepted the default settings for Steps 1 through 6. Scroll down the page and click **Step 7: Map modules to application servers**.
5. On the Step 7: Map modules to application servers page, complete these tasks:
   a. For Module, select **ITSOBankEJB**.
   b. For Clusters and Servers, select **WebSphere:cell=wpslx1,node=wpslx1,server=ITSOBankServer**.
   c. Click **Apply**.
6. Click **Next** to advance to Step 8.
7. Map the ITSOManager role to the `cn=ITSOManagers,cn=groups,dc=itso,dc=ibm,dc=com` group.
   a. For Step 8: Map security roles to users/groups, complete these tasks:
      i. For Role, select **ITSOManager**.
      ii. Click **Lookup Groups**.
   b. On the Lookup users/groups page, complete these tasks:
      i. For Search string, type `ITSOManagers`.
      ii. Click **Search**.
   c. From the Available list, select **cn=ITSOManagers,cn=groups,dc=itso,dc=ibm,dc=com**, and then click **>>** to add it to the Selected list. Click **OK**.
8. Map the ITSOCustomer role to the `cn=ITSOCustomers,cn=groups,dc=itso,dc=ibm,dc=com` group.
   a. For Step 8: Map security roles to users/groups, complete these tasks:
      i. For Role, select **ITSOCustomer**.
      ii. Click **Lookup Groups**.
   b. On the Lookup users/groups page, complete these tasks:
      i. For Search string, type `ITSOCustomers`.
      ii. Click **Search**.
   c. From the Available list, select **cn=ITSOCustomers,cn=groups,dc=itso,dc=ibm,dc=com**, and then click **>>** to add it to the Selected list. Click **OK**.
9. Click **Step 11 Summary page**, and then click **Finish**.
10. Click **Save**.
11. On the Save to Master Configuration page, click **Save**.

12. Restart the ITSOBankServer application server from a console window as follows (restart it if it is already started):

```
# cd /opt/WebSphere/AppServer/bin
# ./stopServer.sh ITSOBankServer -user wpsbind -password wpsbind
# ./startServer.sh ITSOBankServer
```

This completes the ITSO Bank back-end application deployment.

4.3 Deploying the ITSO Bank portal application

This section includes the following tasks to deploy the ITSO Bank portal application:

1. Downloading and unpacking the ITSO Bank sample code
2. Modifying the properties files and repackaging the WAR file
3. Modifying the wmmLDAPServerAttributes.xml file
4. Installing portlets
5. Creating portal pages
6. Adding portlets to pages
7. Modifying resource permissions
8. Verifying the ITSO Bank application
9. Externalizing the ITSO Bank resources

4.3.1 Downloading and unpacking the ITSO Bank sample code

This section explains how to download the ITSO sample code 9121code.zip, unpack the zip file, and extract files from the EAR.

1. Download the ITSO sample code 9121code.zip file from:

   ftp://www.redbooks.ibm.com/redbooks/REDP9121

   > **Note:** For a description of the ITSO sample code and where to download it, refer to Appendix C, “Additional material” on page 221.

2. Unpack the 9121code.zip file. For example, we unpacked the zip to the /tmp/9121code directory.
3. Change the privileges on the files:

   ```
   # chmod -R 777 /tmp/9121code
   ```

4. Copy the ITSOBankWAR.war file to the /tmp/ITSOBankApp directory.

5. Extract the contents of ITSOBankWAR.war to /tmp/ITSOBankApp/ITSOBankWAR.
   a. Open a console window and create the following directory:

      ```
      # mkdir -p /tmp/ITSOBankApp/ITSOBankWAR
      ```

   b. Change to the /tmp/ITSOBankApp/ITSOBankWAR directory.
   c. Enter the following command to extract the ITSOBankWAR.war file:

      ```
      /opt/WebSphere/AppServer/java/bin/jar -xvf ../ITSOBankWAR.war
      ```

   d. Change the privileges on the files:

      ```
      # chmod -R 777 /tmp/ITSOBankApp
      ```

4.3.2 Modifying the properties files and repackaging the WAR file

To deploy the ITSO Bank WAR (ITSOBankWAR.war), perform the following steps:

1. Navigate to the /tmp/ITSOBankApp/ITSOBankWAR/WEB-INF/classes directory where you unpacked ITSOBankWAR.war.
2. Modify the itsobank.properties file.
   a. Modify the following values in the itsobank.properties file for that of the target system (see Example 4-3). The values to change are in bold:
      - **host**: Change the host name to match the host name of the node where the application will be deployed. In our example, this is the Portal Server node (wpslx1).
      - **port**: Change the port to the bootstrap value of the application server that you want to deploy the application to. To determine this value, refer to “Determining an application server bootstrap port” on page 167 (for example, 2811).
      - **managerjndi**: You need to update the node name (for example, `wpwin1`) and application server (for example, ITSOBankServer), as shown in Example 4-3.

Example 4-3 ITSO Bank application itsobank.properties file

```
# ITSO Bank Properties

com.ibm.itso.bank.appserver.host=iiop://wpslx1.itso.ral.ibm.com
com.ibm.itso.bank.appserver.port=2811
com.ibm.itso.bank.appserver.managerjndi=cell/nodes/wpslx1/servers/ITSOBankServer/ITSO/Manager
```

> **Note:** ITSOBankServer is the application server on which this enterprise application is deployed. The ITSOBankServer was created in 4.2.1, “Creating an application server” on page 167. If you choose to change the name of the application server, it should be reflected in the itsobank.properties file.

b. Save and close the itsobank.properties file.

3. Modify the ldaphelper.properties file.

   a. Modify the following values in the ldaphelper.properties file for that of the target system (see Example 4-4). The values to change are highlighted in bold:

      - **java.naming.provider.url**: Host name and port of the Tivoli Directory Server.
      - **java.naming.security.principal**: LDAP administrator ID with write privilege for the parent_dn and parent_dn_admin.
      - **java.naming.security.credentials**: Password for the principal
      - **parent_dn**: Base DN of the LDAP tree for the customer user IDs
      - **parent_dn_admin**: Parent distinguished name (dn) defined during suffix creation

Example 4-4 ITSO Bank application ldaphelper.properties file

```
# The following must be set when name arguments are DN based.

# Use one of the following LDAP servers...
# IBM Directory Server 5.2
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory

# Authentication mechanism. One of none, simple, EXTERNAL or CRAM-MD5.
java.naming.security.authentication=simple

# URL settings. If a Secure Socket Layer (SSL) connection is desired
# then change URL "ldap:" to "ldaps:".

java.naming.provider.url=ldap://ldaplx1.itso.ral.ibm.com:389
java.naming.security.principal=cn=root
java.naming.security.credentials=password

# Parent dn for all user IDs with CUSTOMER role
parent_dn=cn=users,dc=itso,dc=ibm,dc=com

# DN for the ITSOCustomers group
group_dn=cn=ITSOCustomers,cn=groups,dc=itso,dc=ibm,dc=com
```

```
# Parent dn for all user IDs with MANAGER role
parent_dn_admin=cn=users,dc=itso,dc=ibm,dc=com

# Object class required for add user
object_class=top person organizationalPerson ePerson inetOrgPerson
```

b. Save and close the ldaphelper.properties file.

4. Modify the tamhelper.properties file.

   a. Modify the values in the tamhelper.properties file for that of the target system (see Example 4-5). The values to change are in bold.

   > **Note:** The PdPerm.properties file is generated as part of the Tivoli Access Manager Java Runtime Environment configuration (pdjrtecfg).

Example 4-5 ITSO Bank application tamhelper.properties file

```
application=ITSOBank
url=file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
principal=sec_master
credential=password

# Parent dn for all user ID's with CUSTOMER role
parent_dn=cn=users,dc=itso,dc=ibm,dc=com
```

   b. Save and close the tamhelper.properties file.

5. Repackage the ITSOBankWAR.war file.

   a. Copy the original ITSOBankWAR.war file found in the /tmp/ITSOBankApp directory to ITSOBankWAR.war.org.

   b. Navigate to the /tmp/ITSOBankApp/ITSOBankWAR directory.

   c. Repackage ITSOBankWAR.war to include the modified properties files by entering the following command:

```
/opt/WebSphere/AppServer/java/bin/jar -uvf ../ITSOBankWAR.war
WEB-INF/classes/itsobank.properties
WEB-INF/classes/ldaphelper.properties
WEB-INF/classes/tamhelper.properties
```

4.3.3 Modifying the wmmLDAPServerAttributes.xml file

The WebSphere Portal Member Manager maps attribute names that are exposed on Java objects that represent users and groups to the LDAP repository attribute names. Member Manager attributes are mapped to LDAP attributes through the wmmLDAPServerAttributes.xml file.

Note: For more information about the WebSphere Portal Member Manager, refer to the WebSphere Portal InfoCenter at:

`http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html`

Search for `wmmLDAPServerAttributes.xml`.

To modify the wmmLDAPServerAttributes.xml file for the ITSO Bank application example, use the following steps:

1. Navigate to the /opt/WebSphere/PortalServer/wmm directory on the Portal Server node.
2. Back up the wmmLDAPServerAttributes.xml file to wmmLDAPServerAttributes.xml.org.
3. Open the wmmLDAPServerAttributes.xml file.
4. Insert the section listed in Example 4-6 toward the end of the wmmLDAPServerAttributes.xml file (before </repositoryAttributes>, which ends the file). The ITSOid is a correlation ID between the ITSO Bank portlet application and the user data in the LDAP repository.

   You can find the contents of Example 4-6 in /tmp/9121code/config/wps/wmmLDAPServerAttributes_itso.xml.

   Note: In the ITSO Bank application example, we added the `ITSOid` attribute to the LDAP schema in 4.2.4, “Adding the ITSOid attribute to the LDAP schema” on page 170.

   This file is overwritten for certain **`wpsconfig`** tasks such as disable-security and enable-security-ldap. In that case, this entry must be added again.

Example 4-6 ITSO modified wmmLDAPServerAttributes.xml

```
      <attributeMap wmmAttributeName="ITSOid"
            pluginAttributeName="ITSOid"
            applicableMemberTypes="Person"
            dataType="String"
            valueLength="254"
            multiValued="true" />

</repositoryAttributes>
```

5. Save and close the wmmLDAPServerAttributes.xml file.
6. Restart the WebSphere_Portal application server.

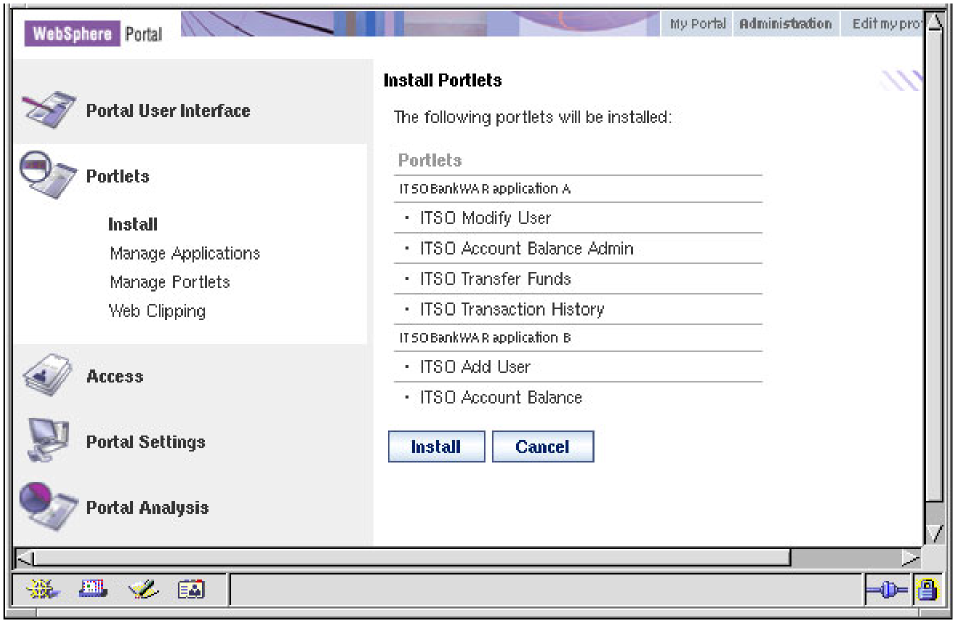
4.3.4 Installing portlets

This section explains how to install the ITSO Bank portlets on the Portal Server node.

1. Access the WebSphere Portal Home page by entering the following URL in a Web browser:

   `https://wslx1.itso.ral.ibm.com/portal/wps/myportal`

2. Log in as user ID `wpsadmin`.
3. Click **Administration**.
4. Click **Portlets →Install**.
5. On the Install Portlets page, specify the location of the file as `/tmp/ITSOBankApp/ITSOBankWAR.war`. Click **Next**.
6. You should see the Install Portlets page like the example in Figure 4-1 that lists the portlets to be installed. Click **Install**.



Install Portlets

The following portlets will be installed:

Portlets
ITSOBankWAR application A
· ITSO Modify User
· ITSO Account Balance Admin
· ITSO Transfer Funds
· ITSO Transaction History
ITSOBankWAR application B
· ITSO Add User
· ITSO Account Balance

Figure 4-1 ITSO Bank portlets to be installed

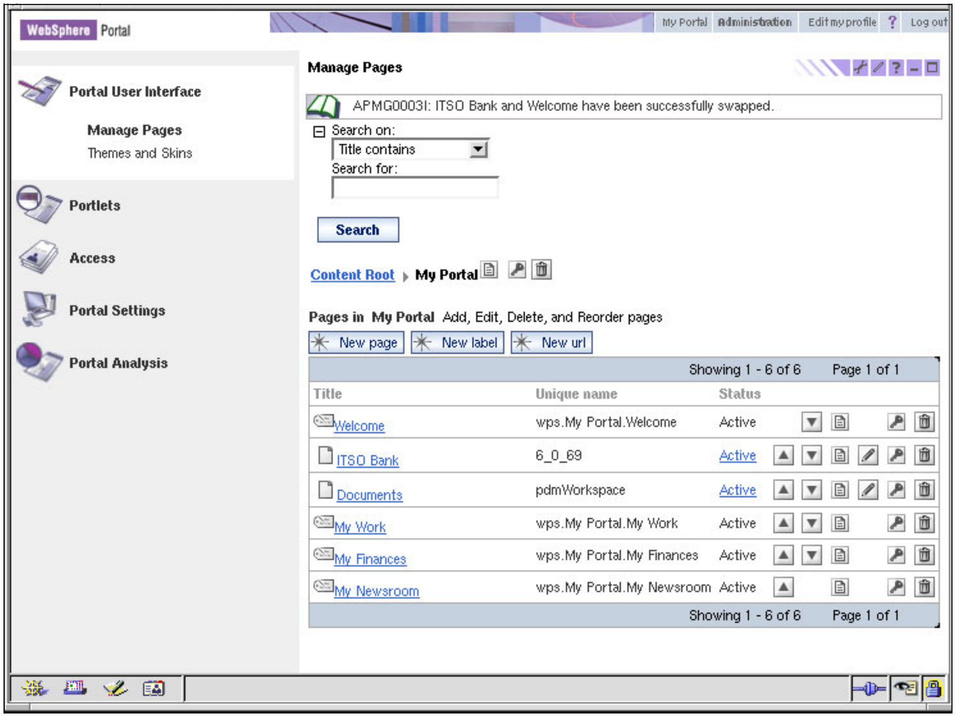
When the installation completes, you should see a message that the portlets were successfully installed.

4.3.5 Creating portal pages

For the ITSO Bank example, we created a hierarchy where the ITSO Bank page acts as a page group that contains three pages (Add User, Modify User, and Access Account).

To create portal pages, perform the following steps:

1. On the Administration page, click **Portal User Interface →Manage Pages**.
2. Create the ITSO Bank page.
   a. On the Manage Pages page, click **My Portal**.
   b. Click **New page**.
   c. On the New page: My Portal page, complete these tasks:
      i. For Title, type `ITSO Bank`.
      ii. For Theme, select **Inherit Parent Theme** (default).
      iii. Click **OK**.
   d. You should see the message `ITSO Bank has been created successfully`. Click **OK**.
3. We want the ITSO Bank page to be displayed after the Welcome page. By default, when added, it is last in the list of pages. Click the up arrow for ITSO Bank several times to move it up the list (before the Welcome page). When you are done, you should see a page like the example in Figure 4-2.



APMG0003I: ITSO Bank and Welcome have been successfully swapped.

Search on: Title contains

Search for:

Pages in My Portal Add, Edit, Delete, and Reorder pages

Showing 1 - 6 of 6 Page 1 of 1

Title	Unique name	Status
Welcome	wps.My Portal.Welcome	Active
ITSO Bank	6_0_69	Active
Documents	pdmWorkspace	Active
My Work	wps.My Portal.My Work	Active
My Finances	wps.My Portal.My Finances	Active
My Newsroom	wps.My Portal.My Newsroom	Active

Showing 1 - 6 of 6 Page 1 of 1

Figure 4-2 Move ITSOBank page

4. Create an Add User page.

   a. Click the **ITSO Bank** page.

   b. Click **New page**.

   c. On the New page: ITSO Bank page, for Title, type `Add User`. Click **OK**.

   d. You should see the message `Add User has been created successfully`. Click **OK**.

5. Create a Modify User page.

   a. Click **New page**.

   b. On the New page: ITSO Bank page, for Title, type `Modify User` and then click **OK**.

   c. You should see the message `Modify User has been created successfully`. Click **OK**.

6. Create Access Account page.

   a. Click **New page**.

   b. On the New page: ITSO Bank page, for Title, type `Access Account` and then click **OK**.

   c. You should see a message `Access Account has been created successfully`. Click **OK**.

4.3.6 Adding portlets to pages

This section explains how to add the ITSO Bank portlets to the pages created in 4.3.5, “Creating portal pages” on page 183.

Adding a portlet to the Add User page

The Add User page should contain the ITSO Add User portlet. This administration portlet allows the manager to enter the details of a new customer. It provides an integrated interface to create a checking and savings account. The ITSO Bank back-end application generates unique account numbers and ITSOid. The customers’ data will be saved in the database and the user will be created in the LDAP with an optional attribute ITSOid set with the unique value.

Add the portlet to the Add User page, as follows:

1. On the Administration page, click **Portal User Interface →Manage Pages**.
2. Navigate to **Content Root →My Portal →ITSOBank**.
3. Click **Add User** page.
4. Click the **pencil** icon (Edit Page Layout).
5. On the Edit Layout page, click the **one column** icon.
6. When you see the message `If you have derived or personalized pages based off this page, then all those changes will be lost. Do you wish to continue?`, click **OK**.
7. Click **Add portlets**.
8. On the Edit Layout page, complete these tasks:
   a. For Search on, select **Title contains**.
   b. For Search for, type `ITSO`.
   c. Click **Search**.
9. When the search results are displayed, you should see a list of ITSO portlets. Select the **ITSO Add User** portlet and then click **OK**.
10. Click **Done**.

Adding portlets to the Modify User page

The Modify User page should contain three portlets including *ITSO Modify User*, *ITSO Transaction History*, and *ITSO Account Balance Admin*. These administration portlets provide extended administrative functionality to the user with the manager role.

The *ITSO Modify User* portlet provides an integrated interface similar to the *ITSO Add User* portlet, which is used to modify the personal details of the chosen customer and the accounts held by the customer. This portlet also incorporates a Click-To-Action icon that enables portlet messaging between the three administration portlets placed on this page. Currently WebSphere Portal Server's property broker can deliver the information through the Click-To-Action technology to the target portlets only when placed on the same page. For this reason, you must place the three administration portlets on the same page.

The portlet ITSO Transaction History allows the manager to view the details of transactions executed on an account or customer. The portlet ITSO Account Balance Admin allows the manager to view the details of the accounts held by the selected customer.

To add the portlets to the Modify User page, follow these steps:

1. Navigate to **Content Root** →**My Portal** →**ITSOBank**.
2. Click **Modify User**.
3. Click the **pencil** icon (Edit Page Layout).
4. For the Modify User page, we use the default two-column page layout. Click **Add portlets** in the left-column container.
5. On the Edit Layout page, complete these tasks:
   a. For Search on, select **Title contains**.
   b. For Search for, type `ITSO`.
   c. Click **Search**.
6. When the search results are displayed, you should see a list of ITSO portlets. Complete the following tasks:
   a. Select **ITSO Account Balance Admin**.
   b. Select **ITSO Modify User**.
   c. Select **ITSO Transaction History**.
   d. Click **OK**.

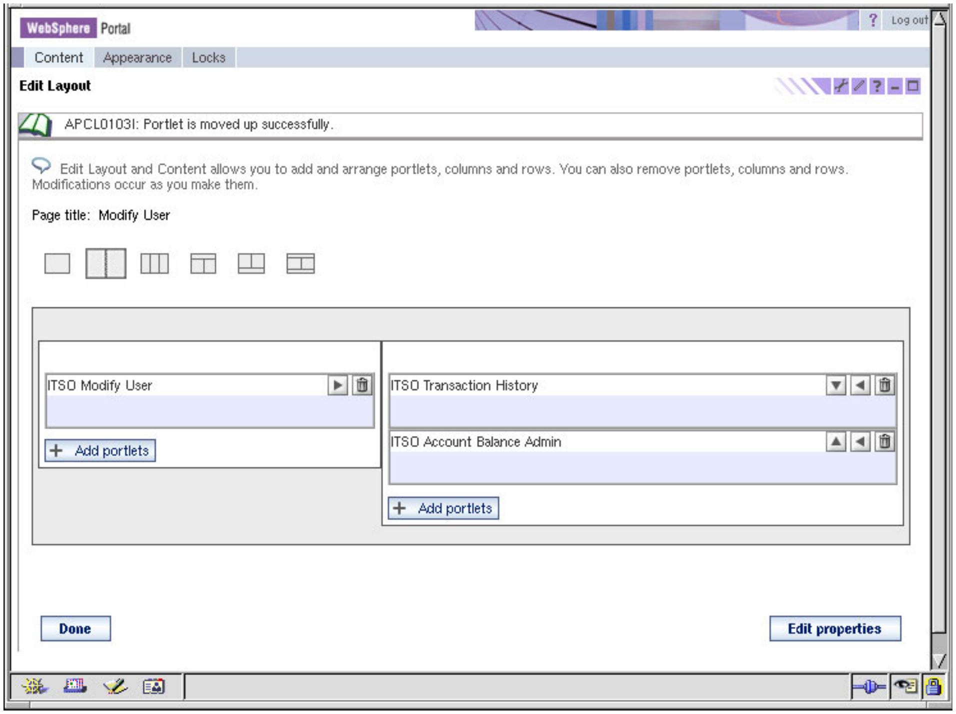
7. Rearrange the portlet layout on the Modify User page.

   a. After you add the portlets to the page, you see a page with all portlets in the left column. Click the right arrow icon (**>**) to move the ITSO Account Balance Admin portlet to the right-column container.

   b. Click the right arrow icon (**>**) to move the ITSO Transaction History portlet to the right-column container.

   c. From the right-column container, click the up arrow (^) for the ITSO Transaction History portlet.

      The portlets should be arranged on the page as shown in Figure 4-3.



APCL0103I: Portlet is moved up successfully.

Edit Layout and Content allows you to add and arrange portlets, columns and rows. You can also remove portlets, columns and rows. Modifications occur as you make them.

Page title: Modify User

Figure 4-3 Modifying the User portlet page layout

8. Click **Done**.

Adding portlets to Access Accounts page

The Access Accounts page should contain two portlets, namely, *ITSO Transfer Funds* and *ITSO Account Balance*. These portlets provide personalized service to the logged in customer. The ITSO Transfer Funds portlet provides a service to transfer funds between the checking and savings accounts. The portlet ITSO

Account Balance provides a service to view the balance in the checking and savings account.

Add the portlets to the Access Account page as follows:

1. Navigate to **Content Root** →**My Portal** →**ITSOBank**.
2. Click **Access Account**.
3. Click the **pencil** icon (Edit Page Layout).
4. On the Edit Layout page, click the **one-column** icon.
5. When you see the message `If you have derived or personalized pages based off this page, then all those changes will be lost. Do you wish to continue?`, click **OK**.
6. Click **Add portlets**.
7. On the Edit Layout page, complete these tasks:
   a. For Search on, select **Title contains**.
   b. For Search for, type `ITSO`.
   c. Click **Search**.
8. When the search results are displayed, you should see a list of ITSO portlets. Complete these tasks:
   a. Select **ITSO Account Balance**.
   b. Select **ITSO Transfer Funds**.
   c. Click **OK**.
9. Rearrange the portlet layout on the Access Account page by clicking the down arrow icon to move the ITSO Account Balance portlet below the ITSO Transfer Funds portlet.
10. Click **Done**.

4.3.7 Modifying resource permissions

This section explains how to set resource permissions for the ITSO Bank pages and portlets based on the roles of users.

Modifying page permissions

The Add User and Modify User pages contain the ITSO Bank application's administration portlets. Only the user ID with the manager role should be granted explicit access to these pages. In our example, we create the user ID manager with the role manager in 4.2.5, "Creating the groups and users for the ITSO Bank application" on page 171. The default behavior of WebSphere Portal Server V5.0.2 is to grant access to the *All authenticated portal users* group for the newly created pages.

To change the page permission so that user IDs with the customer role do not have access permissions to the Add User and Modify User pages, perform the following steps:

1. On the Administration page, click **Access** →**Resource Permissions**.
2. On the Resource Types page, click **Pages**.
3. Click **Content Root**.
4. Click **My Portal**.
5. Assign access to the ITSO Bank page.
   a. Click the **key** icon under Assign access for ITSO Bank page.
   b. Deselect **Allow Inheritance** for the *Privileged User* role and *User* role.
   c. Click **OK**.
6. Click the **ITSO Bank** page.
7. Assign access to the Add User page.
   a. Under Assign access for Add User page, click the **key** icon.
   b. Deselect **Allow Inheritance** for the *Privileged User* role and *User* role.
   c. In the Edit Role column for the Privileged User, click the **pencil** icon.
   d. From the Assign Access for: Pages →Add User page, click **Add**.
   e. On the Search for Users and User Groups page, complete these tasks:
      i. From the Search for Users or User Groups drop-down list, select **User Groups**.
      ii. From the Search on drop-down list, select **cn**.
      iii. In the Search for text box, type `ITSOManagers`.
      iv. Click **Search**.
   f. The page is now updated. In the Select column, select the **ITSOManagers** group check box.
   g. Click **OK**.
   h. Click **Done**.
   i. Click **OK**.
8. Assign access to the Modify User page.
   a. Under Assign access for Modify User page, click the **key** icon.
   b. Deselect **Allow Inheritance** for the *Privileged User* role and *User* role.
   c. In the Edit Role column for the Privileged User, click the **pencil** icon.
   d. From the Assign Access for: Pages →Modify User page, click **Add**.

   e. On the Search for Users and User Groups page, complete these tasks:
      i. From the Search for Users or User Groups drop-down list, select **User Groups**.
      ii. From the Search on drop-down list, select **cn**.
      iii. In the Search for text box, type `ITSOManagers`.
      iv. Click **Search**.
   f. The page is now updated. In the Select column, select the **ITSOManagers** group check box.
   g. Click **OK**.
   h. Click **Done**.
   i. Click **OK**.

9. Assign access to the Access Account page.
   a. Under Assign access for Access Account page, click the **key** icon.
   b. Deselect **Allow Inheritance** for the *Privileged User* role and *User* role.
   c. In the Edit Role column for the Privileged User, click the **pencil** icon.
   d. From the Assign Access for: Pages →Access Account page, click **Add**.
   e. On the Search for Users and User Groups page, complete these tasks:
      i. From the Search for Users or User Groups drop-down list, select **User Groups**.
      ii. From the Search on drop-down list, select **cn**.
      iii. In the Search for text box, type `ITSOCustomers`.
      iv. Click **Search**.
   f. The page is now updated. In the Select column, select the **ITSOCustomers** group check box.
   g. Click **OK**.
   h. Click **Done**.
   i. Click **OK**.
   j. Click **Done**.

Modifying portlet permissions

The default behavior of WebSphere Portal V5.0.2 is to disable access to portal users for the newly installed portlets. This section explains how to assign permissions for the ITSOManagers and ITSOCustomers groups for the listed portlets:

- ITSOManagers group

  The ITSO Bank portlets used by the ITSOManagers group are:

  - ITSO Account Balance Admin
  - ITSO Add User
  - ITSO Modify User
  - ITSO Transaction History

- ITSOCustomers group

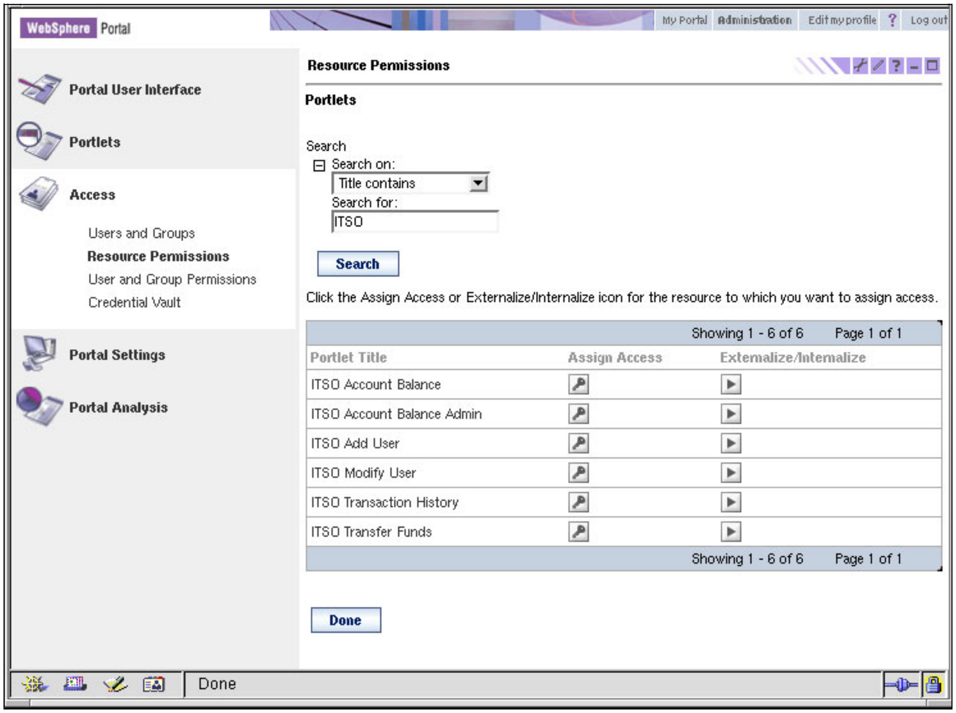
  The ITSO Bank portlets used by the ITSOCustomers group are:

  - ITSO Account Balance
  - ITSO Transfer Funds

To modify the portlet permissions, follow these steps:

1. On the Administration page, click **Access** →**Resource Permissions**.
2. Click **Portlets**.
3. On the Portlets Search page, complete these tasks:
   a. For Search on, select **Title contains**.
   b. For Search for, type `ITSO`.
   c. Click **Search**.

From the results of the search, you should have a list of ITSO portlets, as shown in Figure 4-4.



Resource Permissions

Portlets

Search

Search on: Title contains

Search for: ITSO

Search

Click the Assign Access or Externalize/Internalize icon for the resource to which you want to assign access.

Showing 1 - 6 of 6 Page 1 of 1

Portlet Title	Assign Access	Externalize/Internalize
ITSO Account Balance	🔑	▶
ITSO Account Balance Admin	🔑	▶
ITSO Add User	🔑	▶
ITSO Modify User	🔑	▶
ITSO Transaction History	🔑	▶
ITSO Transfer Funds	🔑	▶

Showing 1 - 6 of 6 Page 1 of 1

Figure 4-4 ITSO Bank portlets

Modifying portlet permissions for the ITSOManagers group

To modify the portlet permissions for the ITSOManagers group, use the following steps for each of these portlets:

- ITSO Account Balance Admin
- ITSO Modify User
- ITSO Transaction History
- ITSO Add User

1. Modify the ITSO Account Balance Admin portlet permissions for the ITSOManagers group:

   a. Under Assign access, click the **key** icon for *ITSO Account Balance Admin* portlet.

   b. Deselect **Allow Inheritance** for the *User* role.

   c. In the Edit Role column, click the **pencil** icon for the *Privileged User*.

   d. On the Assign: Portlets →ITSO Account Balance Admin page, click **Add**.

e. On the Search for Users and User Groups page, complete these tasks:

   i. From the Search for Users or User Groups drop-down list, select **User Groups**.

   ii. From the Search on drop-down list, select **cn**.

   iii. In the Search for text box, enter `ITSOManagers`.

   iv. Click **Search**.

f. The page is now updated. In the Select column, select the **ITSOManagers** group check box.

g. Click **OK**.

h. Click **Done**.

i. Click **OK**.

2. Repeat the process to modify portlet permissions (step 1) for each of the remaining portlets for the ITSOManagers group.

   – ITSO Modify User
   – ITSO Transaction History

> **Note:** The ITSO Modify User portlet, ITSO Transaction History portlet, and ITSO Account Balance portlet are developed to incorporate Click-To-Action technology supported by WebSphere Portal's Property Broker. They are programmatically enhanced to support the creation and deletion of wire between the Click-To-Action enabled portlets. This administration of wire requires the role of Privileged User. Therefore, the ITSOManagers group is granted the role of Privileged User only for these three portlets.

3. Modify the ITSO Add User portlet permissions for the ITSOManagers group:

   a. Under Assign access, click the **key** icon for ITSO Add User portlet.

   b. Deselect **Allow Inheritance** for the *User* role.

   c. In the Edit Role column, click the **pencil** icon for the *User*.

   d. On the Assign: Portlets →ITSO Add User page, click **Add**.

   e. On the Search for Users and User Groups page, follow these steps:

      i. In the Edit Role column, select **User Groups**.

      ii. From the Search on drop-down list, select **cn**.

      iii. In the Search for text box, type `ITSOManagers`.

      iv. Click **Search**.

   f. The page is now updated. In the Select column, select the **ITSOManagers** group check box.

g. Click **OK**.

h. Click **Done**.

i. Click **OK**.

Modifying portlet permissions for the ITSOCustomers group

To modify the portlet permissions for the ITSOCustomers group, perform the following steps for each of these portlets:

- ITSO Account Balance
- ITSO Transfer Funds

1. Modify the ITSO Account Balance portlet permissions for the ITSOCustomers group:

   a. Under Assign access, click the **key** icon for ITSO Account Balance portlet.

   b. Deselect **Allow Inheritance** for the *User* role.

   c. In the Edit Role column, click the **pencil** icon for the *User*.

   d. On the Assign: Portlets →ITSO Account Balance page, click **Add**.

   e. On the Search for Users and User Groups page, complete these tasks:

      i. From the Search for Users or User Groups drop-down list, select **User Groups**.

      ii. From the Search on drop-down list, select **cn**.

      iii. In the Search for text box, type `ITSOCustomers`.

      iv. Click **Search**.

   f. The page is now updated. In the Select column, select the **ITSOCustomers** group.

   g. Click **OK**.

   h. Click **Done**.

   i. Click **OK**.

2. Repeat the process to modify portlet permissions (step 1) for the *ITSO Transfer Funds* portlet for the ITSOCustomers group.

3. Log out of WebSphere Portal.

4.3.8 Verifying the ITSO Bank application

This section explains how to perform a basic test to verify that the ITSO Bank application is working properly after deployment and setting permissions within WebSphere Portal.

1. Enter the following URL in a Web browser to access the WebSphere Portal home page:

   ```
   https://wslx1.itso.ral.ibm.com/portal/wps/myportal
   ```

2. Log in as user ID `manager1`.

   The manager1 user ID was created in 4.2.5, “Creating the groups and users for the ITSO Bank application” on page 171.

3. Click the **ITSOBank** tab.
4. Click **Add User**.

> **Note:** If the Add User portlet fails with the error message `The portlet was not available. Please contact your portal administrator`, you may have missed the step to restart the ITSOBankServer. Restart the ITSOBankServer for changes to take effect.

4.3.9 Externalizing the ITSO Bank resources

This section explains how to externalize resources for the ITSO Bank portlet pages. It includes the following steps:

1. ITSO Bank externalizing a resource overview
2. Backing up systems prior to externalizing a resource
3. Externalizing the ITSO Bank pages resource
4. Verifying the Tivoli Access Manager object space and ACLs.
5. Verifying the ITSO Bank application

ITSO Bank externalizing a resource overview

When externalizing the ITSO Bank page resources, we need to explicitly define the permission assignments for the pages due to the inheritance being severed. Table 4-1 defines the ITSO Bank page resource permission assignments defined within WebSphere Portal.

Table 4-1 ITSO Bank page resource permission assignments

Resource types		**Roles**	
ITSO Bank pages		**Privileged User**	**Administrator**
ITSO Bank		None	wpsadmins
	Add User	ITSOManagers	wpsadmins
	Modify User	ITSOManagers	wpsadmins
	Access Account	ITSOCustomers	wpsadmins

Table 4-2 defines the ITSO Bank portlet resource permission assignments defined within WebSphere Portal.

Table 4-2 ITSO Bank portlet resource permission assignments

Resource types	**Roles**	
ITSO portlets	**User**	**Administrator**
ITSO Account Balance Admin	ITSOManagers	wpsadmins
ITSO Add User	ITSOManagers	wpsadmins
ITSO Modify User	ITSOManagers	wpsadmins
ITSO Transaction History	ITSOManagers	wpsadmins
ITSO Account Balance	ITSOCustomers	wpsadmins
ITSO Transfer Funds	ITSOCustomers	wpsadmins

Backing up systems prior to externalizing a resource

We recommend that you perform system backups prior to externalizing your resources. Within the ITSO example runtime environment, we performed a backup of the Portal Server node and the Policy Server node.

Externalizing the ITSO Bank pages resource

This section explains how to externalize the ITSO Bank pages resource using the WebSphere Portal Resource Permission portlet.

1. Enter the following URL to start the WebSphere Portal Administration Console:

   ```
   https://wslx1.itso.ral.ibm.com/portal/wps/myportal
   ```

2. Log on as the `wpsadmin` user.
3. Access the Resource Permission portlet. Click **Administration** →**Access** → **Resource Permissions**.
4. Externalize the resource permissions for the ITSO Bank page.
   a. Under Resource Type column, click **Pages**.
   b. In the Page Title column, click **Content Root**.
   c. Click **My Portal**.
   d. In the Externalize/Internalize column for the ITSO Bank page, click the right arrow (>).
   e. You are prompted with the message `Are you sure you want to place the resource under External Access Control?` Click **OK**.

   f. You should see the message `Resource Successfully externalized.` Click **Done**.

5. Externalize the resource permissions for portlets.

   a. Click **Portlets**.

   b. From the Portlet search page, complete these tasks:

      i. For Search on, select **Title contains**.
      ii. For Search for, type `ITSO`.
      iii. Click **Search**.

   c. Repeat the following steps for each of these portlets:

      - ITSO Account Balance
      - ITSO Account Balance Admin
      - ITSO Add User
      - ITSO Modify User
      - ITSO Transaction History
      - ITSO Transfer Funds

      i. In the Externalize/Internalize column for the respective portlet, click the right arrow (>).

      ii. You are prompted with the message `Are you sure you want to place the resource under External Access Control?` Click **OK**.

   d. After you externalize all the ITSO portlets, click **Done**.

Verifying the Tivoli Access Manager object space and ACLs

Verify that the objects have been created properly in the Tivoli Access Manager object space and access control lists (ACLs) have been created. You can perform the verification by using the Tivoli Access Manager **`pdadmin`** command line tool or the Web Portal Manager Web-based interface.

Verifying the ITSO Bank application

Now that the ITSO Bank pages are externalized and the roles are assigned, verify the access to the ITSO Bank pages.

A

SUSE LINUX tips

This appendix highlights the key steps that we performed to install SUSE LINUX Enterprise Server V8 for the ITSO working example secure portal runtime environment. In addition, this appendix includes details about performing common administrative tasks and examples for executing commands.

Installing SUSE LINUX

This section explains the steps that we used to install SUSE LINUX Enterprise Server V8 for the ITSO working example. This procedure includes post installation configuration and the installation of SUSE LINUX Service Pack 3.

The SUSE LINUX installation entails the following tasks:

1. Planning considerations
2. Installing SUSE LINUX Enterprise Server V8
3. Configuring the SUSE LINUX post installation
4. Installing SUSE LINUX V8 Service Pack 3
5. Installing and updating Perl
6. Configuring IBM Java Runtime Environment V1.3.1
7. Installing Mozilla Web browser (optional)
8. Starting VMWare toolbox (optional)

Planning considerations

Both IBM WebSphere Portal Extend for Multiplatforms V5.0.2 and IBM Tivoli Access Manager for e-business V5.1 support SUSE LINUX Enterprise Server V8 and Red Hat Enterprise Linux V2.1. In this redpaper, we test only the configuration with SUSE LINUX. The SUSE LINUX Enterprise Server V8 is built upon the UnitedLinux V1.0 distribution.

Tivoli Directory Server and openldap-client

The openldap-client libraries are installed by default during the SUSE LINUX Enterprise Server V8 installation. The Tivoli Directory Server installer overwrites some of the openldap-client libraries. There are couple of possible methods to resolve this issues.

- Uninstall openldap-client files.

  Uninstall the openldap-client if it is not needed.

- Move openldap-client files to avoid configuration conflict and keep shared libraries dependent on the openldap-client needed by many other Linux packages.

> **Note:** We chose this second option for the redpaper. This option moves the openldap-client files to a new directory (/usr/bin/openldapclient) prior to the Tivoli Directory Server. It allows the shared libraries that are dependent on the openldap-client to exist on the node. It also provides an environment that does not conflict with the Tivoli Directory Server.

- Coexistence between openldap-client and Tivoli Directory Server.

  For more information about this option, refer to the article on the Web at:

  http://www.ibm.com/developerworks/linux/library/l-ss4-itds/

Java Runtime Environment

By default, SUSE LINUX installs the SUN Java Runtime Environment (JRE) V1.3. We found that such applications as DB2 Universal Database (UDB), Tivoli Access Manager, and the IBM GSKit were dependent on the IBM Java Runtime V1.3.1. We found that the SUN JRE caused problems to leave it installed. For this reason, we uninstalled the SUN JRE.

For more information, see “Configuring IBM Java Runtime Environment V1.3.1” on page 207.

Perl required by IBM GSKit installer

Perl is required by the IBM GSKit installer. We need an updated version of the IBM GSKit to properly create the key store and certificates. The version of the IBM GSKit supplied with the IBM Tivoli Directory Server V5.2 does not work properly.

Refer to “Installing and updating Perl” on page 206 for details.

VMWare Workstation tools considerations

If you are using VMWare to build a test environment, as is the case for the ITSO working example test environment, after the SUSE LINUX installation, you must install the VMWare tools to optimize performance.

Refer to “Installing VMWare tools” on page 204 for more information.

VMWare Workstation and network configuration

If you are using VMWare to build a test environment, as is the case for the ITSO working example test environment, after the SUSE LINUX installation, we recommend that you configure the TCP/IP network using static addresses and use the /etc/hosts file to resolve names. For example, we included the following in /etc/hosts file for each of our nodes:

```
192.168.2.100 ldaplx1.itso.ral.ibm.com ldaplx1
192.168.2.110 wpslx1.itso.ral.ibm.com wpslx1
192.168.2.120 tamlx1.itso.ral.ibm.com tamlx1
192.168.2.130 wslx1.itso.ral.ibm.com wslx1
```

Installing SUSE LINUX Enterprise Server V8

1. Insert *UnitedLinux CD1* and boot it from the drive to start the installer.
2. When you see the SUSE LINUX Enterprise Server startup window, select **Installation**.
3. In the UnitedLinux license agreement window, review the agreement and click **Accept** if you agree to the conditions of the agreement.
4. In the Welcome to YaST window, select the desired language (for example, English - US) and then click **Accept**.
5. In the Installation Settings window, make the appropriate selections. Then click **Accept**. For this example, we made the following selections:
   - Mode: **new installation** (default)
   - Keyboard layout: **English - US** (default)
   - Mouse: **IntelliWheel mouse - Aux-port** (default)
   - Partitioning:
     - Create swap partition 1.0 GB on /dev/sda1
     - Create root partition 9.0 GB (/dev/sda2 with reiser)

     > **Note:** In our example, we use VMWare Workstation V4.5.2 with a 10 GB virtual hard disk 1024 MB of virtual memory. For a test environment, we choose to create one root partition for all software to be installed. In a production environment, you may choose to create separate partitions.

   - Software:

     There are many possible software options. We selected the following options for our configuration by clicking **Software** and the clicking **Detailed selection**. Complete the following tasks:

     i. C/C++ Compiler and Tools:
     – Select **gcc**, **gcc-c++**, **kernel-source**, **make**
     – Deselect **gdb**.

     ii. Deselect **Gnome system**.
     iii. Select **KDE Desktop Environment** (default).
     iv. Deselect **Simple Webserver**.
     v. Select **LSB Runtime Environment** (default).
     vi. Select **Help & Support Documentation** (default).
     vii. Select **Graphical Base System** (default).
     viii.Select **YaST2 config modules**.
     ix. Select **Analyzing Tools**.
     x. Deselect **Authentication Server - NIS, LDAP, Kerberos**.

xi. Deselect **DHCP and DNS Server**.
xii. Select **File & Print Server - NFS, Sambak, Cups**.
xiii. Deselect **Mail and News Services**.
xiv. Select **SLES Administration Tools**.
xv. Click **Accept**.

- Booting: Booting from 1.SCSI 10 GB, /dev/sda, VMWare, -VMWare Virtual S (default)
- Time zone: **US/Eastern**
- Language: **English - US**

6. When you are finished, click **Accept**.
7. You may see the warning message `YaST2 has obtained all the information required to install SUSE LINUX. The installation will be carried out according to settings made in the previous dialogs. To commit the installation and all choices mad so far, choose “Yes”. Choose “No” to return to the previous dialog. Start Installation?` Click **Yes, install**.
8. Insert *UnitedLinux Version 1.0 CD1* when prompted and then click **OK** to continue.
9. When the installation is complete, you should see the message `The base system was successfully installed. Your machine must now be rebooted. Please remove all installation media (CD-ROM)`. Click **OK**.
10. During the installation, insert *UnitedLinux Version 1.0 CD2* when prompted. Click **OK** to continue.
11. During the installation, insert *SuSe SLES Version 8 CD1* when prompted. Click **OK** to continue.
12. You should see a message `The base system was successfully installed. Your machine must now be rebooted. Please remove all installation media (CD-ROM)`. Click **OK** to proceed with the system restart.
13. After the system restarts, enter a password for the root user when prompted and then click **Next**.
14. In the Add a new user window, create a user (for example, admin) and then click **Next**.
15. Select **Graphical desktop environment** and click **Accept**.
16. In the Network Interfaces window, click **Yes** to autodetect. Click **Next**.
17. The system restarts. Then log on as a root user.

Configuring the SUSE LINUX post installation

This section explains how to configure SUSE LINUX after you install the base SUSE LINUX for the ITSO working example runtime environment.

Installing VMWare tools

If you have installed Linux in a VMware image, you need to install the VMWare tools to optimize performance of the video and other drivers. You cannot simply install VMWare tools from the menu. You need to copy the tools from the iso image, unpack using tar and run the installation as follows:

1. Set the VMWare hot keys to Crtl+Shift+Alt.

   By default, VMWare uses Ctrl+Alt, which is the same key sequence used by Linux to invoke the Text (Ctrl+Alt, F1) and Graphical (Ctrl+Alt, F7) modes. Switching to the Text mode is necessary to install the VMWare tools. To resolve this conflict, set the VMWare hot keys to Crtl+Shift+Alt as follows:

   a. From the VMWare menu, select **Edit** →**Preferences**.
   b. Click the **Hot keys** tab.
   c. Select the **Ctrl+Shift+Alt** radio button (default is Ctrl+Alt) and click **OK**.

   > **Note:** Remember that VMWare now uses Ctrl+Shift+Alt for its hot key sequence in place of Ctrl+Alt.

2. From the VMWare menu, select **VM** →**Install VMWare Tools**. Click **Install**.
3. Press Ctrl+Alt and then F1 to switch to the Linux Text mode in preparation to install the tools.

   > **Note:** The VMWare tools installation invokes the X86Free utility to set the video resolution for Linux. X86Free requires that the program run from Text mode.
   >
   > - To switch to Text mode, press Ctrl+Alt and then F1.
   > - To switch to Graphics mode, press Ctrl+Alt and then F7.

4. Enter the following command from the Text mode command line:

```
mount /dev/cdrom /mnt
cd /mnt
cp vmware-linux-tools.tar.gz /tmp
cd /tmp
gunzip vmware-linux-tools.tar.gz
tar -xvf vmware-linux-tools.tar
cd vmware-tools-distrib
./vmware-install.pl
```

5. During the installation of the VMWare Tools, you are prompted for additional information. During the tools installation, the X86Free utility is run to set your video resolution (for example, we used 1024x768).
6. Press Ctrl+Alt and then F7 to switch back to Graphics mode.
7. Select **VM** →**Cancel VMWare Tools Install** since we already installed the tools.

For more information about installing the VMWare tools on Linux, see the VMWare Web site at:

`http://www.vmware.com`

TCP/IP network configuration

Refer to “Configuring TCP/IP network” on page 210 for details.

Installing SUSE LINUX V8 Service Pack 3

There are several methods to install SUSE LINUX V8 Service Pack 3 (download CD image, FTP, HTTP, etc.). In our example, we download the service pack 3 CD images. For more information about installing service pack 3, see:

`http://www.suse.com`

To install SUSE LINUX V8 Service Pack 3 from the CD images, follow these steps:

1. Click **Start Application** →**System** →**YaST2** to launch YaST2.
2. Double-click **Patch CD Update**.
3. Select **Automatic Update**, accept the default **cd:///** for the Installation Source, and click **Next**. This installs the online update patches in preparation for further updates.
4. When you see the message `Please restart the online update to get all available patches`, click **OK**.
5. Now you see the message `Installation successful`. Click **OK**.
6. In the Online Update Confirmation window, click **Finish**.
7. Double-click **Patch CD Update**.
8. Select **Automatic Update**, accept the default **cd:///** for the Installation Source, and click **Next**. This installs the Recommended updates, security patches, and Service Pack 3.
9. You are prompted regarding lilo with instructions on how to proceed. Click **OK** to continue.

10. When you see the message Please restart your vncserver processes, click **OK** to continue.

11. When you see the message, reboot the system with the shutdown -r now to load the new kernel, click **OK** to continue.

12. When you see the message, For the openssh update package to be to become effective, run the command “rcsshd restart” as root, click **OK**.

13. You then see another lilo message. Click **OK**.

14. At the end of the installation summary, you should see the message, Service Pack 3 and installation finished. Click **Next**.

15. Click **Finish**.

16. Click **Close** to close YaST.

17. Restart the system.

> **VMWare tools:** After you install SUSE LINUX V8 Service Pack 3, ensure the VMWare tools are still active and installed. If your VMWare session displays a message on the bottom of the window that the tools are not installed, reinstall the VMWare tools as described in “Installing VMWare tools” on page 204.

Installing and updating Perl

After installing SUSE LINUX V8 Service Pack 3, we found that the Perl package has some inconsistencies. Specifically, the updated perl-XML-DOM rpm had a dependency on perl-XML-RegExp. perl-XML-RegExp is not included in the base SUSE LINUX Enterprise Server V8 distribution CDs, but it is included on the SUSE LINUX V8 Service Pack 3 CD1.

Perl is required by the IBM GSKit installer, which is required for the ITSO working example runtime environment. To install Perl with the perl-XML-RegExp rpm, perform the following steps:

1. To determine if Perl is installed, enter the following command:

   ```
   # rpm -qa | grep perl-XML
   ```

   If the base Perl rpms are installed, you should see the Perl modules listed.

   ```
   perl-XML-Writer-0.4-254
   perl-XML-Generator-0.9-164
   perl-XML-DOM-1.39-95
   ```

   > **Note:** The updated perl-XML-DOM-1-39-95 has a dependency on the perl-XML-RexExp rpm, which is not installed.

If the base Perl modules are not installed, use YaST to install the Perl package.

2. To update Perl with the perl-XML-RegExp rpm, complete these tasks:
   a. Insert *SUSE LINUX V8 Service Pack 3 CD1*.
   b. Open a terminal window as a root user and mount the CD-ROM:
      ```
      # mount /dev/cdrom /mnt
      ```
   c. Change to the following location on the CD-ROM (/mnt) where the perl-XML-RegExp rpm is located:
      ```
      # cd /mnt/UnitedLinux/i586
      ```
   d. Install the perl-XML-RegExp rpm:
      ```
      # rpm -U --nodeps perl-XML-RegExp-0.03-380.i586.rpm
      ```
   e. Verify Perl rpms are installed, by entering the following command:
      ```
      # rpm -qa | grep perl-XML
      ```
      If the Perl rpms are installed, you should see the following Perl modules listed:
      ```
      perl-XML-Writer-0.4-254
      perl-XML-Generator-0.9-164
      perl-XML-RegExp-0.03-380
      perl-XML-DOM-1.39-95
      ```
3. Click **Start Application** →**System** →**YaST2** to launch YaST2.
4. Double-click **Install or remove software**.
5. From the Filter drop-down list, select **Package Groups**.
6. Select **Developer** →**Libraries** →**Perl**.
7. Click **Accept**.
8. When prompted, insert *UnitedLinux Version 1.0 CD1* and then click **OK**.
9. Close YaST2.

Configuring IBM Java Runtime Environment V1.3.1

Several of the application used in the secure portal solution require a JRE (for example, DB2 UDB, IBM GSKit, Tivoli Access Manager PDJRTE, etc.). By default, the SUSE LINUX Enterprise Server V8 installation includes several Java packages and runtime environments, which are not compatible versions with the IBM applications that use the JRE.

There are a few possible solutions to this problem. For example, you can symbolically link the IBM JRE in place of the java2-jre. For simplicity on a production environment, we chose to uninstall the java2-jre and export the JAVA_HOME configured to the IBM JRE.

Table A-1 lists the Java packages that are installed by the default SUSE LINUX V8 installation.

Table A-1 Java packages installed with SUSE LINUX V8

Package name	Version	Description
IBMJava2-JAAS	1.3.1-24	Java Authentication and Authorization Service (JAAS) V1.0 for Linux
IBMJava2-JAVACOMM	1.3.1-24	Java Communications API for Linux V2.01
IBMJava2-JRE	1.3.1-24	IBM Runtime Environment for Linux, Java 2 Technology Edition, Version 1.3.0
IBMJava2-SDK	1.3.1-24	IBM Developer Kit for Linux, Java 2 Technology Edition
java2-jre	1.3.1-524	Java 2 Runtime Environment

The following procedure explains how to uninstall the java2-jre-1.3.1-524 and set the JAVA_HOME to the IBM JRE.

1. Open a console window as a root user.
2. To uninstall java2-jre-1.3.1-524 included with SUSE LINUX, complete these tasks:

   a. To determine if the SUN JRE V1.3.1-524 is installed, enter the following command in a console window:

   ```
   # rpm -qa | grep java2-jre
   ```

   If the SUN JRE is installed, you should see the following output:

   ```
   java2-jre-1.3.1-524
   ```

   b. Click **Start Application** →**System** →**YaST2** to launch YaST2.

   c. Double-click **Install or remove software**.

   d. From the Filter drop-down list, select **Package Groups**.

   e. Select **Developer** →**Languages** →**Java**.

   f. Deselect **java2-jre**.

   g. Click **Accept**.

   h. Close YaST2.

3. Determine if the IBM JRE V1.3.1-24 is installed by entering the following command:

   ```
   # rpm -qa | grep IBMJava2
   ```

   If the JRE is installed, you should see the following output:

   ```
   IBMJava2-JAAS-1.3.1-24
   IBMJava2-JAVACOMM-1.3.1-24
   IBMJava2-JRE-1.3.1-24
   IBMJava2-SDK-1.3.1-24
   ```

4. Modify the JAVA_HOME variable in /etc/profile.local as follows:

   > **Note:** The /etc/profile is a global file. We recommend that you do not update this file directly, but rather update profile.local.

   a. Open a shell console and navigate to the /etc directory:

   ```
   # cd /etc
   ```

   b. Open /etc/profile.local in an editor to add the variable:

   ```
   # kate profile
   ```

   c. Add the following lines to the profile.local file:

   ```
   JAVA_HOME=/usr/lib/IBMJava2-1.3.1
   export JAVA_HOME
   PATH=$PATH:$JAVA_HOME/bin
   export PATH
   ```

   d. Save and close the file.

5. Verify that the JRE is configured properly.

   a. Enter the following command to display the JAVA_HOME:

   ```
   # $JAVA_HOME
   ```

   You should see the following output:

   ```
   bash: /usr/lib/IBMJava2-1.3.1: is a directory
   ```

   b. Enter the following command to test that the JRE bin is in the path:

   ```
   # java -version
   ```

   You should see the following output:

   ```
   Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1)
   Classic VM (build 1.3.1, J2RE 1.3.1 IBM build cxia32131-20020622 (JIT
   enabled: jitc))
   ```

Installing Mozilla Web browser (optional)

Installing the Mozilla Web browser is optional. We use Mozilla instead of the built-in SUSE LINUX Konqueror Web browser. For details about installing Mozilla for Linux, refer to the following Web site:

http://www.mozilla.org/products/mozilla1.x/

By default the Mozilla Web browser is installed to the /usr/local/mozilla directory and can be started with the following command:

```
./mozilla
```

Note: As an alternative to the Mozilla Web browser, you may consider Firefox, which you can find on the Web at:

http://www.mozilla.org/products/firefox/

Starting VMWare toolbox (optional)

To run the VMWare toolbox in Linux for such tasks as shrinking a disk, use these steps:

1. Open a console window as a root user.
2. Start the vmware toolbox. Enter the following command:

```
# vmware-toolbox
```

Configuring the network

This section explains how the following procedures:

- Configuring TCP/IP network
- Configuring Samba

Configuring TCP/IP network

You can configure the TCP/IP network settings by updating the configuration files manually or using the YaST graphical user interface (GUI).

Configuring DHCP TCP/IP

When using DHCP, you are required only to define the host name and domain. The DHCP client automatically retrieves the proper DNS and default router. Alternatively, these values can be manually set when using DHCP.

By default, after you install SUSE LINUX, the system TCP/IP network is configured to use DHCP. The /etc/sysconfig/network/ifcfg-eth0 file is configured for DHCP as shown in Example A-1 to retrieve the network information.

Example: A-1 Sample ifcfg-eth0 configuration for DHCP

```
STARTMODE="onboot"
BOOTPROTO="dhcp"
```

To configure the host name and domain for DHCP using YaST, follow these steps:

1. Click **Start Application** →**System** →**YaST2** to launch YaST2.
2. Select **Network/Basic** →**Network card configuration**.
3. Select the already configured network adapter and click **Change**.
4. For the eth0 device, click **Edit**.
5. Select **Automatic address setup (via DHCP)**.
6. Click **Host name and name server**.
7. Click **Accept** to update name server and search list via DHCP.
8. In the Host name and DNS window, enter the following values as we did in our example:
   a. For Host name, type `sles8`.
   b. For Domain name, type `itso.ral.ibm.com`.
   c. Deselect **Change host name via DHCP**.
   d. Click **Finish**.
9. Click **Next**.
10. Click **Next** to continue.
11. Click **Finish**.
12. Click **Close** to close YaST.

Configuring static IP TCP/IP

To configure Linux to use a static IP address using YaST, use these steps:

1. Click **Start Application** →**System** →**YaST2** to launch YaST2.
2. Select **Network/Basic** →**Network card configuration**.
3. Select the already configured network adapter and click **Change**.
4. For the eth0 device, click **Edit**.

5. Select **Static address setup**. Then enter the following values:
   a. IP address: `192.168.2.100`
   b. Subnet mask: `255.255.255.0`
6. Click **Host name and name server**.
7. In the Host name and DNS window, enter the following values as we did for this example:
   a. For Host name, type `sles8`.
   b. For Domain name, type `itso.ral.ibm.com`.
   c. For Name server, type `192.168.2.1`.
   d. For Domain search, type 1: `itso.ral.ibm.com`.
   e. Click **Next**.
8. Click **Routing**.
9. In the Routing window, for the Default gateway, enter an IP address. We entered `192.168.2.1`. Then click **Next**.
10. Click **Next** to continue.
11. Click **Finish**.
12. Click **Close** to close YaST.
13. Verify the configuration.

    The YaST tool updates the following files:

    – Modifies the /etc/sysconfig/network/ifcfg-eth0 for your environment to retrieve the network information statically (see Example A-2)

Example: A-2 Sample ifcfg-eth0 configuration for static IP address

```
STARTMODE=‘onboot‘
BOOTPROTO=static’
IPADDR=192.168.2.100
NETMASK=255.255.255.0
BROADCAST=192.168.2.255
```

– Modifies the /etc/resolv.conf file

This file defines the Domain Name Server (DNS), domain, and the domain search order; for example, out /etc/resolv.conf file contains:

```
search itso.ral.ibm.com
nameserver 192.168.2.1
```

– Adds the routing configuration to /etc/sysconfig/network/routes

```
default 192.168.2.1
```

– At the end of the YaST Network card configuration, runs the **`SuSEconfig`** command

Verifying the network

This section explains how to verify that the Linux system networking is working properly.

1. Verify /etc/hosts file.

   If you do not have a host entry for the host name of your node, add it. For example, our /etc/hosts file contains:

   ```
   127.0.0.1 localhost.localdomain localhost
   192.168.2.100 sles8.itso.ral.ibm.com sles8
   ```

2. Verify the /etc/resolv.conf file.

   This file defines the Domain Name Server (DNS), domain, and the domain search order. For example, our /etc/resolv.conf file contains:

   ```
   search itso.ral.ibm.com
   nameserver 192.168.2.1
   ```

3. Verify the network configuration using the **ifconfig** command. The **ifconfig** command returns:

   ```
   # ifconfig
   ```

4. Verify the route:

   ```
   # route -n
   ```

   Verify that you can access other systems and your system by name (resolves name with DNS or hosts file).

Configuring Samba

Samba uses the Server Message Block (SMB) protocol based on NetBIOS over TCP/IP. Samba can be used to provide file and print sharing that is compatible with Microsoft Windows systems.

This section explains the basic steps for configuring a Samba server and configuring a client to access Samba files shares on Linux systems as well as Windows file shares. The samba package is required to be installed on SUSE LINUX for both the server and client.

Configuring the Samba server

In this section, we explain how to configure a Samba file share. The /etc/samba/smb.conf file is used to define file and printer shares as shown in Example A-3. The smb.conf file is organized into two sections, global and shares. The Samba server can be configured manually or by using the swat GUI interface.

For more information about Samba, see the Samba Web site at:

http://www.samba.org

Example: A-3 Sample smb.conf

```
# ###################################################################
# Global parameters
[global]
   encrypt passwords = Yes
   map to guest = Bad User
   time server = Yes
   unix extensions = Yes
   socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
   printcap name = CUPS
   os level = 2
   printing = cups
   veto files = /*.eml/*.nws/riched20.dll/*.{*}/

# ###################################################################

# Home directories
[homes]
   comment = Home Directories
   valid users = %S
   read only = No
   create mask = 0640
   directory mask = 0750
   browseable = No

# ###################################################################
# File share
[software]
   comment = Secure Portal for Linux software
   path = /home/resident/software
```

To configure the Samba server and create a file share using the swat GUI, follow these steps:

Note: For illustration purposes, we create a file share named *software* in the /home/resident/software directory owned by user resident. We used this file share as a repository for all the application software needed to be installed in the ITSO working example runtime environment.

We found that the WebSphere Portal Installer failed when installing over the Samba network share. We recommend that you copy the images locally, unpack them, and then install them from the local drive.

1. Create a Linux user named admin using YaST.
   a. Click **Start Application** →**System** →**YaST2** to launch YaST2.
   b. Select **Security and Users** →**Edit and create users**.
   c. Click **Add**.
   d. In the Add a new user window, complete the following steps as we did for this example:
      i. For First name, type `Redbook`.
      ii. For Last name, type `Resident`.
      iii. For User login, type `resident`.
      iv. Enter a password.
      v. Re-enter the password for verification.

      > **Note**: If you want to change the default groups, login shell, etc., click **Details**.

      vi. Click **Create**.
   e. Click **Finish**.
   f. Close YaST.
2. Enable the swat service.
   a. Click **Start Application** →**System** →**YaST2** launch YaST2.
   b. Select **Network/Basic** →**Start/stop services (inetd)**.
   c. In the Network services window, select **On with custom configuration** and then click **Next**.
   d. Scroll down the list, select **swat** and then click **Activate or Deactivate** (toggle). The swat status should now be *Active*.
   e. Click **Finish**.
3. Access the swat GUI from a Web browser to configure the Samba server.
   a. Enter the following URL in a Web Browser:

      ```
      http://hostname:901
      ```
   b. Enter a user ID and password when prompted.
4. Create a file share from the swat GUI.
   a. From the swat home page, click **Shares**.
   b. In the Create share text field, enter the sharename (for example, `software`) and then click **Create share**.

   c. On the Share Parameters page, enter the values for Comment and Path. We entered the following values for sw share:

      - Comment: `Secure Portal for Linux software`
      - Path: `/home/resident/software`

      Note: We accepted the defaults for the remaining configuration options.

      Click **Commit Changes**.

5. Create Samba users using the swat GUI.
   a. From the swat home page, click **Password**.
   b. Under Server management, enter the username and password that you want to create and then click **Add New User** (for example, we entered the resident user and password).
   c. In the User name field, enter the desired user name, click **Enable user**.
6. Ensure that the Samba services are running.
   a. From the swat home page, click **Status**.
   b. Click **Start smbd** and **Start nmbd** if the server status is not already running.

The resident user can now access the software file share.

Configuring the Samba client

The Samba client allow you to connect from a Linux system to Windows or other Linux systems Samba server file and printer shares.

For example, you may want to mount a Windows 2000 Server file share (`websphere` is the server, `sw` is the share) from your Linux server (Samba client) to access software during the installation process. We include two methods to mount a Samba file share. The first method is:

```
# mount -t smbfs -o username=WindowsUserID,password=WindowsUserPassword
//<WindowsMachineName>/WindowsShare /DirectoryOnLinuxServer
```

For example, we would enter:

```
# mount -t smbfs -o username=admin,password=password //websphere/sw /mnt
```

The second method is:

```
# smbmount /WindowsMachineName/WindowsShare /DirectoryOnLinuxServer -o
username=WindowsUsername,password=WindowsPW
```

Common commands

This section highlights common Linux commands and tasks:

- Shutdown
  - To shut down now without a file system check, enter:

    ```
    shutdown -f now
    ```

  - To shut down and reboot without a file system check, enter:

    ```
    shutdown -f -r now
    ```

- Mount a CD-ROM

  ```
  mount /dev/cdrom /media/cdrom
  ```

- Unmount a CD-ROM

  ```
  unmount /media/cdrom
  ```

- Add users

  ```
  adduser<username
  ```

- Add groups

  ```
  groupadd groupname
  ```

- Change a users password

  ```
  passwd user id
  ```

- Monitor disk space and file system usage

  ```
  df
  du
  ```

- Directory listing

  ```
  ls -l (long form list)
  ls -al (hidden files)
  ls (short form list)
  ```

- Showing the processes running

  ```
  ps -axf
  ps -axf -width=240
  ps -ef
  ```

- Finding a file

  ```
  locate file name
  find target directory -name file name -print
  ```

- Determining the network IP address

  ```
  ifconfig -a
  ```

- Network restart commands **ifdown** and **ifup**

  ```
  ifdown eth0 (bring ethernet interface down)
  ifup eth0 (bring ethernet interface up)
  ```

- Utility to check, add, or delete a service in /etc/init.d

  ```
  chkconfig
  ```

- tar command
  - Extract files:

    ```
    tar -xvf filname.tar
    ```

  - Compress files:

    ```
    tar -cvf * filname.tar
    ```

- gunzip command

  ```
  gunzip -d -c filename.gz
  ```

  - Variation with *filename*.tar.gz

    ```
    gunzip -d -c filename.tar.gz | tar -xvf -
    ```

  - Variation with *filename*.taz

    ```
    gunzip -d -c filename.taz | tar -xvf -
    ```

- unzip command

  ```
  unzip filname.zip
  ```

- vmstat - check memory usage

  ```
  vmstat
  ```

B

Security hardening

After the base installation and configuration of the runtime environment are complete, the environment should be hardened. For example, you need to configure the application middleware and connections between the nodes for security and SSL enabled. This topic is beyond the scope of this redpaper. We recommend that you reference the following chapters of the *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325 redook:

- Security domain and risk management
- Security hardening

Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1, SG24-6325, also includes a sample application that you can use to deploy.

C

Additional material

This Redpaper refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redpaper is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/REDP9121`

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 Redpaper form number, REDP9121.

Using the Web material

The additional Web material that accompanies this Redpaper includes the following files:

File name	*Description*
9121code.zip	Working example sample configuration scripts zipped.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	5 MB
Operating System:	Linux
Processor:	1 GHz
Memory:	1024 MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder (for example, c:\9121code).

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 226. Note that some of the documents referenced here may be available in softcopy only.

- *Understanding LDAP - Design and Implementation*, SG24-4986
- *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855
- *IBM Host Integration in a Secure Network*, SG24-5988
- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- *IBM WebSphere Portal V5, A Guide for Portlet Application Development*, SG24-6076
- *A Secure Portal Using WebSphere Portal V5 and Tivoli Access Manager V4.1*, SG24-6077
- *A Portal Composite Pattern Using WebSphere Portal V5*, SG24-6087
- *IBM WebSphere Portal for Multiplatforms V5 Handbook*, SG24-6098
- *IBM WebSphere Application Server V5.0 Systems Management and Configuration: WebSphere Handbook Series*, SG24-6195
- *WebSphere V5.0 Applications: Ensuring High Performance and Scalability*, SG24-6198
- *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325
- *IBM WebSphere V5.0 Security, WebSphere Handbook Series*, SG24-6573
- *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885
- *WebSphere Studio Application Developer Version 5 Programming Guide*, SG24-6957
- *Identity Management Design Guide with IBM Tivoli Identity Manager*, SG24-6996

- *Lotus Security Handbook*, SG24-7017
- *IBM Tivoli Access Manager for e-business*, REDP-3677
- *A Secure Portal Extended With Single Sign-On*, REDP-3743

Other publications

These publications are also relevant as further information sources:

- *WebSphere Portal V5.0.2.1 with Tivoli Access Manager V5.1.0.2 Integration Guide, Volume 1: Installation and Configuration* white paper by Ray Neucom

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101710
- Stuart McClure, et al. *Hacking Exposed: Network Security Secrets & Solutions*. Osborne/McGraw Hill, 2001. ISBN 0072193816.
- *Installation and Configuration Guide, IBM Tivoli Directory Server V5.2*, SC32-1338
- *Administration Guide, IBM Tivoli Directory Server V5.2*, SC32-1339
- *Performance Tuning Guide, IBM Tivoli Directory Server V5.2*, SC32-1342
- *Authorization Java Classes Developer Reference, IBM Tivoli Access Manager V5.1*, SC32-1350
- *Performance Tuning Guide, IBM Tivoli Access Manager V5.1*, SC32-1351
- *Problem Determination Guide, IBM Tivoli Access Manager V5.1*, SC32-1352
- *Error Message Reference, IBM Tivoli Access Manager V5.1*, SC32-1353
- *Command Reference, IBM Tivoli Access Manager V5.1*, SC32-1354
- *Administration Java Classes Developer Reference, IBM Tivoli Access Manager V5.1*, SC32-1356
- *Web Security Developer Reference, IBM Tivoli Access Manager V5.1*, SC32-1358
- *WebSEAL Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1359
- *Base Administration Guide, IBM Tivoli Access Manager V5.1*, SC32-1360
- *Web Security Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1361
- *Base Installation Guide, IBM Tivoli Access Manager V5.1*, SC32-1362
- *IBM Global Security Kit, Secure Sockets Layer Introduction and iKeyman User's Guide V7a*, SC32-1363

- *Plug-in for Web Servers Integration Guide, IBM Tivoli Access Manager V5.1*, SC32-1365
- *Plug-in for IBM WebSphere Edge Server Integration Guide, IBM Tivoli Access Manager V5.1*, SC32-1367
- *IBM WebSphere Application Server Integration Guide, IBM Tivoli Access Manager V5.1*, SC32-1368

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM WebSphere Portal for Multiplatforms InfoCenter:

  http://www.ibm.com/websphere/portal/library
- IBM WebSphere Portal Extend for Multiplatforms home page

  http://www.ibm.com/software/genservers/portal/
- IBM WebSphere Portal Extend for Multiplatforms technical library home page

  http://www.ibm.com/software/genservers/portal/library/
- IBM Tivoli Access Manager for e-business home page

  http://www.ibm.com/software/tivoli/products/access-mgr-e-bus/
- IBM Tivoli Access Manager for e-business technical information home page

  http://publib.boulder.ibm.com/tividd/td/IBMAccessManagerfore-business5.1.html
- The Center for Internet Security (CIS)

  http://www.cisecurity.org/
- Microsoft Security home page

  http://www.microsoft.com/security/default.asp
- System Administration, Networking and Security Institute (SANS)

  http://www.sans.org/
- SANS Institute, Information Reading Room for Windows 2000

  http://www.sans.org/rr/whitepapers/win2k/
- SANS Institute, Information Reading Room for general UNIX®

  http://www.sans.org/rr/whitepapers/unix/

- IBM AIX® home page

  http://www.ibm.com/servers/aix/

- IBM AIX Service Bulletins and Security Advisories

  http://www-1.ibm.com/servers/eserver/support/pseries/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Deploying a Secure Portal Solution on Linux

Using WebSphere Portal V5.0.2 and Tivoli Access Manager V5.1

Redpaper

Implement a secure portal runtime environment on Linux

Configure authentication, single signon, and authorization

Deploy the ITSO Bank secure portal application on Linux

Portals provide a personalized single point of access to applications, content, and processes through a Web interface. Secure portal solutions are needed to address common security challenges, such as authentication, authorization and single signon.

This IBM Redpaper and the accompanying sample code provide IT architects, IT specialists, and administrators with the critical knowledge to implement the secure portal solution runtime environment and secure an application. The runtime environment includes IBM WebSphere Portal Extend for Multiplatforms V5.0.2.2 and IBM Tivoli Access Manager for e-business V5.1.0.4 on the SUSE LINUX Enterprise Server V8 platform.

For more information about the architecture of the secure portal solution and development related topics, we recommend that you refer to the IBM Redbook *Develop and Deploy a Secure Portal, Using WebSphere Portal V5.0.2 and Tivoli Access Manager 5.1*, SG24-6325.