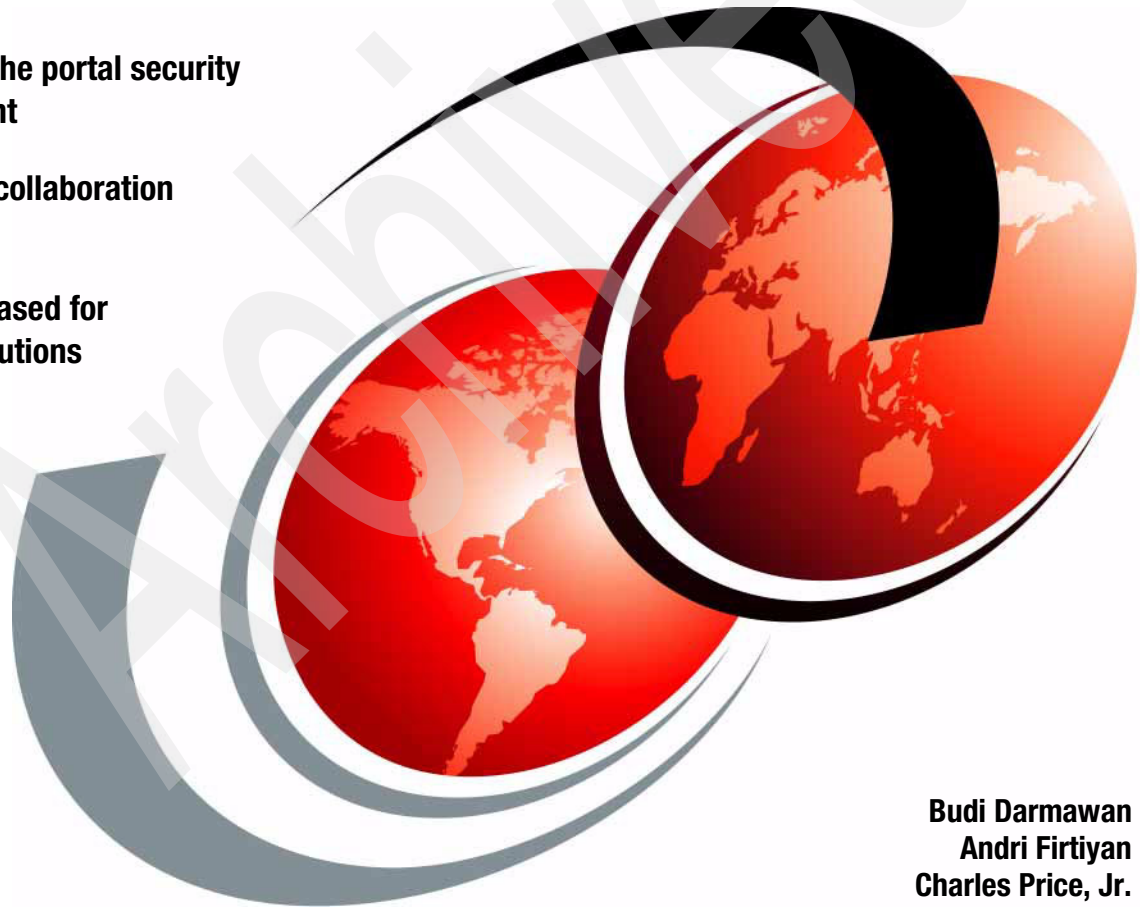IBM

WebSphere Portal Collaboration Security Handbook

Describes the portal security environment

Integrates collaboration solutions

Scenario-based for identity solutions

Budi Darmawan
Andri Firtiyan
Charles Price, Jr.

Redbooks

IBM

International Technical Support Organization

WebSphere Portal Collaboration Security Handbook

December 2004

First Edition (December 2004)

This edition applies to Version 5, Release 0, Modification 2 of IBM WebSphere Portal Extend for Multiplatforms, and IBM Lotus Collaboration Center, IBM Lotus Instant Messaging and Web Conferencing Version 6.5.1, IBM Lotus Domino Server Version 6.5.1, and IBM Lotus Team Workplace Version 6.5.1.

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.

ix

Trademarks

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

AIX®	ibm.com®	Redbooks (logo) ™
Cloudscape™	IBM®	Redbooks™
DB2 Universal Database™	iNotes™	Sametime®
DB2®	Lotus Discovery Server™	Tivoli®
Domino Designer®	Lotus Notes®	WebSphere®
Domino.Doc®	Lotus®	Workplace Messaging™
Domino®	Notes®	Workplace™
e-business on demand™	QuickPlace®	

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.

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

Security is the hottest topic in the current Web-centric computing environment. This issue becomes the single largest concern for IT professionals who are stakeholders for Web applications, such as administrators, programmers, and users.

In this IBM Redbook, we discuss this security issue with the implementation of IBM WebSphere® Portal Extend for Multiplatforms in an IBM Lotus® collaborative environment. This discussion is scenario-based and aims to assist in the deployment of WebSphere Portal with Lotus Collaborative Components in a secure implementation. We describe several degrees of security, noting their advantages and disadvantages.

The primary goal of this scenario is to have a WebSphere Portal server with Lotus Team Workplace™ (formerly called QuickPlace®) and Lotus Instant Messaging and Web Conferencing (formerly called Sametime®) environment set up and running securely.

We discuss proxy authentication with IBM Tivoli® Access Manager for e-business Version 5.1 and discuss the use of various identity providers, such as IBM Tivoli Directory Server, Domino® LDAP, and Microsoft® Active Directory.

The team that wrote this redbook

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

Figure 1   From left to right: Charles Price, Jr., Budi Darmawan, and Andri Firtiyan

Budi Darmawan is a Consulting IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on all areas of systems management. Before joining the ITSO six years ago, Budi worked in IBM Global Services, Indonesia as a Solution Architect and Lead Services Implementer. His current interests include autonomic computing, business service management, Java™ programming, and Web security.

Andri Firtiyan is a Senior Consultant with Softworks Solution, a consulting firm in Jakarta, Indonesia (http://www.softworks.biz). Previously, Andri worked as an IT Specialist for the IBM Software Group, Indonesia for the past four years. He has nine years of experience in software application development. He holds a master's degree in Computer Science from the University of Indonesia. His areas of expertise include Java, object-oriented design analysis, and IBM WebSphere. He is also co-author the *IBM WebSphere Application Server - Express V5.0.2 Developer Handbook*, SG24-6555.

Charles Price, Jr. is a Software Engineer in the IBM Software Group, U.S. He has four years of experience in technical support for IBM Lotus software. He holds a degree in Mathematics Education from the University of Georgia and

taught high school mathematics for three years before joining IBM. His areas of expertise include Lotus Domino administration and the Lotus collaborative portlets. He is an IBM Certified Associate System Administrator - Lotus Collaborative Solutions (administering QuickPlace and administering Sametime) and a Principal Certified Lotus Professional for Domino system administration and is currently working on certification as an IBM Certified System Administrator for WebSphere Portal.

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

William Tworek, Chris Almond, Axel Buecker, John Ganci, Elizabeth Barnes
International Technical Support Organization

Garren Linker
IBM U.S., Tivoli Systems

Stephen Shepherd
IBM U.S., Lotus software

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.co**m/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

> **ibm.com**/redbooks

► Send your comments in an email to:

    redbook@us.ibm.com

► Mail your comments to:

    IBM® Corporation, International Technical Support Organization
    Dept. OSJB Building 905
    11501 Burnet Road
    Austin, Texas 78758-3493

1

Portal security introduction

This redbook describes the security setup of IBM WebSphere Portal and its relation to IBM Lotus collaboration software. This chapter serves as the introduction to this book and also defines some basic concepts that we use later in the book.

In this chapter, we discuss the following topics:

► Security in the on demand world: Shows the role of security in the current on demand operating environment, specifically relating to WebSphere Portal technology.

► Portal security needs: Discusses applicable security issues and configuration related to the WebSphere Portal environment and Lotus collaboration software.

► Overview of IBM products: Gives an overview of all the related software products that are used in this redbook, their relationships, and their positioning related to the secure solution.

► Document structure: Maps the content of this book into suggested reading trails that you can follow.

1.1 Security in the on demand world

Enterprises are now residing in the interconnected world where Web applications are the keyword for accessibility and survival. The interconnected nature of the Web applications results in the emphasis on the importance of security when accessing these applications.

A conceptual Web application interconnection is shown in Figure 1-1.
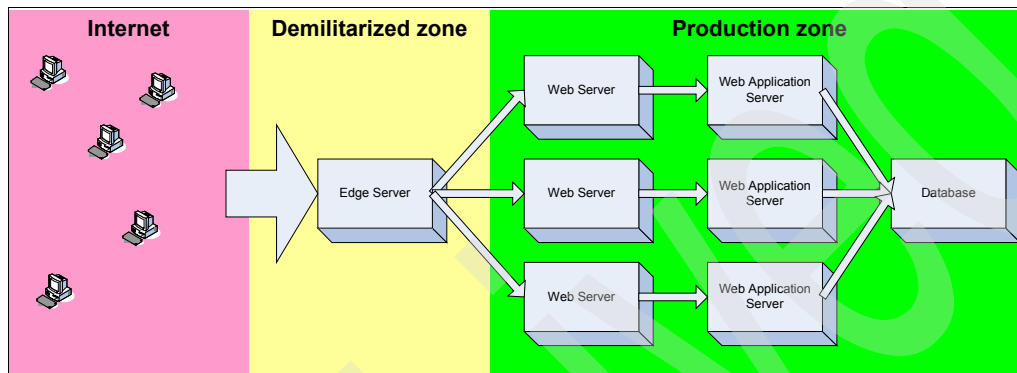
Figure 1-1   Web application interconnection

In this environment, the ability to serve the consumer is critical for the survival of the business. There are several key factors that needs to be considered in this case, such as the availability, performance, usability, and security of the application and servers. This redbook specifically highlights security.

Security is necessary, typically for the following reasons:

► Communications are generally performed through a public network

► Public connectivity invites attempts to get into the network.

► There is a necessity to shield some restricted functionality from the public.

However, the effort to ensure security must also consider that too much security will be a burden to end users and might drive them away from the application.

A secure, on demand Web application with a focus on user interaction is key to the integrity of the application. Web portal technology presents a way to encapsulate multiple applications in a single Web browser, similar to a graphical desktop with multiple active windows. This application model has additional complexity, because the user must be authenticated and authorized for potentially more than one application, and these applications can interact with other applications.

1.2 Portal security needs

In the WebSphere Portal environment, some common security needs have to be addressed to ensure the overall integrity of the application. These security needs can be categorized into the following categories:

► Traffic encryption: To prevent eavesdropping of the communication, all network traffic that goes in and out of a Web site needs to be encrypted. This encryption enables private communication between the user and provider of application to be secured.

► User authentication: To really understand the accessing parties, the user's identity needs to be known and trusted. There are several methods of authenticating users. The basic method is using a user ID and password combination and can be enhanced with other identification methods, such as certificates.

► Function authorization: Some specific functions might need to be specifically protected from general access. This protection can depend on the required granularity of access, such as a specific function or a specific action, by a certain account number, and so on.

► Credential passing: To remove the need for end users to authenticate multiple times while using different functions or applications, and instead to use a seamless user interface. This is especially important in a portal environment, because a single Web portal page can be from several applications, which might require different authentication mechanism. This function is typically called single sign-on (SSO).

► Protocol filtering: To focus on a specific aspect of securing the environment, a mechanism can be employed to only let certain protocols to enter the network. This mechanism is commonly known as a firewall. Firewalls enable system administrator to focus on securing a subset of protocol and to filter out potentially insecure protocols.

► Intrusion detection: To intelligently detect intrusions and attacks to your network and respond with the appropriate action and minimize the impact on security and performance of your systems.

The following sections discuss these security needs in more detail.

1.2.1 Encryption

Typical encryption in Web applications uses the Secure Sockets Layer (SSL) protocol enhancement over Hypertext Transfer Protocol (HTTP) or other protocol communications.

The use of SSL involves encryption. Encryption requires the use of keys that are stored in certificates. This needs to be established encryption takes place. For more information about encryption and certificates, see 2.2, "Communication encryption" on page 15.

1.2.2 Authentication

Authentication is typically performed using an integrated corporate directory or identity management. Some common or well-known identity solutions include:

► Lightweight Directory Access Protocol (LDAP)

► Microsoft Active Directory user registry

► Lotus Domino address book

In this book, we discuss several possible authentication mechanisms integrated with the WebSphere Portal solution.

1.2.3 Authorization

Each user must be associated with a set of functions that the user needs to execute. The user needs to be able to access those functions, and only those functions. Additional restrictions might be needed for the user, such as a qualifying access level to the functions. For example, an end user might need read-only access to display an account balance, while a bank teller has full control to create new accounts.

In the Java 2 Platform, Enterprise Edition (J2EE) security framework, each application or Enterprise JavaBeans can be associated with a role. Each role can be associated with access authorization for its functions and its access levels. These roles can then be assigned to individuals.

1.2.4 Single sign-on

Single sign-on (SSO) is needed to alleviate the burden of authenticating to multiple applications individually. Users should be able to sign on only once and gain access to all the back-end applications. This can be achieved using several mechanisms, such as:

► Passing an LTPA token that indicates the LDAP directory entry

► Sending a stored user ID and password pair to the back-end applications

► Using a certificate file to identify the user

► Using a trusted connection to acquire just the user ID that has already been authenticated

1.2.5 Protocol filtering

Protocol filtering is commonly known as a firewall. This is a mechanism that is put in place to select which requests can be forwarded into a part of the network. Typically, the filtering is performed by a set of rules that define the characteristics of the communication packages that are allowed to get through. The filtering mechanism is performed by comparing the source port, destination port, protocol family, host addresses, and other attributes to the rules.

Figure 1-2 shows a formal division of a network in an enterprise.
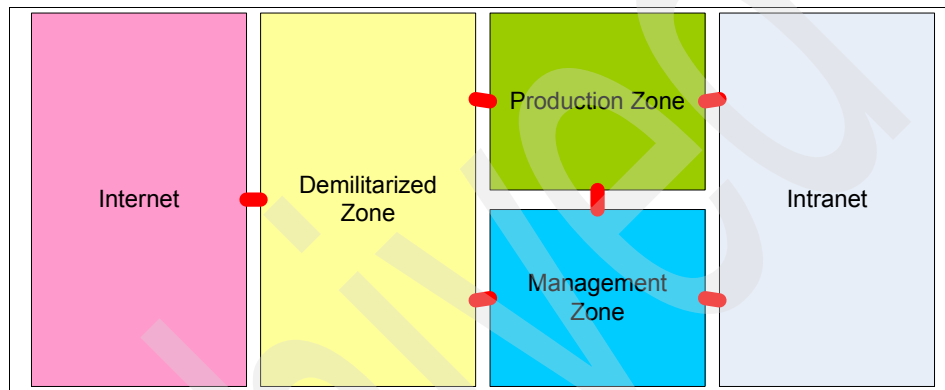
Figure 1-2   Enterprise network configuration

We do not discuss this protocol filtering in this book, because the mechanism is common to all Web applications.

1.2.6 Intrusion detection

Network intrusion is a common situation in the current state of networked environments. People (malicious hackers) try to gain access to network and applications. Some of the attempts are detectable from the network devices, such as an excessive number of malformed packages, a sudden increase of network volume, frequent unauthorized access retries, and other events.

These patterns can be correlated and analyzed to form a solid lead to a possible intrusion, and automated actions can be performed to prevent network outage and defend the network. This intrusion detection system is not in the scope of this book.

1.3 Overview of IBM products

In this redbook, the scenario of securing a WebSphere Portal server relates to the implementation of several different IBM software components. We also discuss some non-IBM software components.

1.3.1 IBM WebSphere Portal Extend for Multiplatforms

This is the IBM e-business on demand™ workplace that provides a robust, portal-based Web browser solution. The technology is built on the award-winning WebSphere Application Server Version 5 platform using J2EE standards to optimize performance.

WebSphere Portal Extend for Multiplatforms delivers a single, universal point of access that is integrated, highly customizable, extensible, and scalable to interact with key applications, content, people, and business processes.

In this book, this product is often called the WebSphere Portal server. The product includes the following components:

► IBM WebSphere Portal

► IBM WebSphere Application Server Enterprise Version 5.0 Fix Pack 1

► IBM Directory Server Version 5.1

► IBM DB2® Universal Database™ Enterprise Server Edition Version 8.1 Fix Pack 1

► IBM WebSphere Portal toolkit

► WebSphere Studio Site Developer Version 5.0 Fix Pack 1

► WebSphere Portal content publishing

► IBM Tivoli Web Site Analyzer Version 4.5

► IBM WebSphere Translation Server Version 5.0

► IBM Lotus Collaborative Components

► IBM Lotus Collaboration Center

► IBM Lotus Extended Search Version 4.0

► Instant messaging and online awareness (based on Lotus Sametime Version 3.0)

► Virtual teamrooms (based on Lotus QuickPlace Version 3.0.1)

1.3.2 IBM Lotus Instant Messaging and Web Conferencing

IBM Lotus Instant Messaging and Web Conferencing (formerly called Sametime) Version 6.5.1 is the IBM platform for real-time collaboration. It is based on three on demand concepts:

- ► Presence awareness: See, in advance, whether a person or persons or an application or applications are available to collaborate, share information, take an action, or all of these.
- ► Instant messaging: Converse virtually through the exchange of text-based, audio-based, and video-based information in real time.
- ► Web conferencing: Share information, an application, or an entire desktop, or engage in team white boarding.

Though basic in nature, these capabilities present customers with virtually unlimited possibilities. Lotus Instant Messaging and Web Conferencing is based on Lotus Domino server technology.

1.3.3 IBM Lotus Team Workplace

IBM Lotus Team Workplace (formerly called QuickPlace) Version 6.5.1 is a business-ready, self-service work space expressly designed for team collaboration. With Lotus Team Workplace, users can instantly create secure work spaces on the Web, providing them with a "place" to coordinate, collaborate, and communicate on any project or ad hoc initiative. Key Lotus Team Workplace capabilities include:

- ► Coordination: People, tasks, plans, and resources
- ► Collaboration: Ideas and discussion, issues, shared documents, files, and general due diligence
- ► Communication: Actions and decisions, key findings and lessons, and knowledge capture

Organizations of all sizes can take advantage Lotus Team Workplace out-of-the-box, or they can easily customize it to meet their unique business requirements. Lotus Team Workplace is based on the Lotus Domino server technology.

1.3.4 IBM Lotus Domino

IBM Lotus Domino Version 6.5.1 provides a multiplatform foundation for collaboration and e-business, driving solutions from corporate messaging to Web based transactions, and everything in between. This enterprise-class messaging

and collaboration system is built to maximize human productivity by unleashing the experience and expertise of individuals, teams, and extended communities.

Initially developed to act as a mail server, Lotus Domino has currently served as a collaborative platform with multiple functionality, such as providing workflow-based applications, document storage, and archival systems, in addition to other possibilities.

1.3.5 IBM Lotus Workplace

IBM Lotus Workplace is an integrated family of collaborative products based on open standards. Lotus Workplace leverages the use of DB2 Universal Database (UDB) and WebSphere Portal technologies. Lotus Workplace combines market-leading collaborative products that can be experienced through a choice of security-rich clients, giving people simplified access and interaction with other people and a host of collaborative capabilities, such as e-mail, calendaring, and scheduling, instant messaging, Web conferencing, team spaces, document and Web content management, and learning.

In addition, Lotus Workplace is enabled to deliver a variety of server-managed client experiences ranging from browser-based to a new full rich client. The server-managed client model provides administrators a security-rich, no touch deployment model coupled with central policy-based management of the end user's desktop environment.

Enhanced Lotus Workplace products are:

► IBM Lotus Workplace Messaging™: Enabled by the new, innovative Workplace Client Technology of IBM, Lotus Workplace Messaging offers a standards-based, simple messaging experience for browser users, while now providing server-managed delivery of a rich client experience for those users who can benefit from an extended set of integrated productivity tools.

► IBM Lotus Workplace Team Collaboration: An integrated instant messaging and presence awareness, Web conferencing, and customizable team spaces product, Lotus Workplace Team Collaboration helps individuals, teams, and entire organizations, together with their customers, Business Partners, and suppliers, increase business efficiency and improve productivity all while managing cost of ownership.

► IBM Lotus Workplace Collaborative Learning: Helps organizations manage their training programs more efficiently and deliver a variety of learning experiences to users. Improved integration with the Lotus Workplace platform increases personal, team, and organizational productivity through collaboration and access to timely, centralized information and online learning resources.

- ► IBM Lotus Workplace Web Content Management: Helps streamline the Web content management process by providing a rapid content deployment capability and point-and-click interface. This helps relieve IT and webmaster bottlenecks by placing content creation and management in the hands of content experts for author once, publish everywhere control.

- ► IBM Lotus Workplace Documents: Enabled by the new, innovative Workplace Client Technology of IBM, Lotus Workplace Documents provides a low-cost, standards-based collaborative document management product with a choice of rich client or browser-based experience that enables the management of the complete life cycle of office documents, from collaborative authoring to review, approval, and archival.

1.3.6 IBM Tivoli Access Manager for e-business

IBM Tivoli Access Manager for e-business is an award winning, policy-based access control solution for e-business and enterprise applications that is in the leader quadrant of Gartner's Magic Quadrant. Tivoli Access Manager for e-business can help you manage growth and complexity, control escalating management costs, and address the difficulties of implementing security policies across a wide range of Web and application resources.

Tivoli Access Manager for e-business integrates with e-business applications out-of-the-box to deliver a secure, unified, and personalized e-business experience. By providing authentication and authorization APIs and integration with application platforms such as J2EE, Tivoli Access Manager for e-business helps you secure access to business-critical applications and data spread across the extended enterprise.

Web-based single sign-on (SSO) can span multiple sites or domains by exploiting the Tivoli Access Manager cross-domain SSO technology or by using Security Assurance Markup Language (SAML) and other token-passing protocols.

1.3.7 IBM Tivoli Directory Server

IBM Tivoli Directory Server provides a powerful Lightweight Directory Access Protocol (LDAP) identity infrastructure that is the foundation for deploying comprehensive identity management applications and advanced software architectures such as Web services.

Tivoli Directory Server is a powerful, security-rich, and standards-compliant enterprise directory for corporate intranets and the Internet. Tivoli Directory Server is built to serve as the identity data foundation for rapid development and

deployment of your Web applications and security and identity management initiatives by including strong management, replication, and security features.

With Tivoli Directory Server, you can choose your authentication strategy: You can use simple user ID and password authentication, or you can implement the more secure digital certificate-based authentication structure. Tivoli Directory Server also includes a Simple Authentication Security Layer (SASL) plug-in interface, including Challenge-Response Authentication Mechanism MD5 (CRAM-MD5) and Kerberos authentication if required.

The fine-grained access control features in Tivoli Directory Server extend to the attribute level, enabling self-service and delegated administration while also offering protection of access control list (ACL) values within the directory, preventing unauthorized users from changing the security assigned to objects within the directory.

Development and deployment of your enterprise directory with Tivoli Directory Server is enhanced through the inclusion of the IBM default schema, a flexible server plug-in framework and the client SDK that includes support for 64-bit IBM AIX® and Java access through a standard J2EE interface.

IBM Tivoli Directory Server is a component of the IBM identity management solution that can help you get users, systems, and applications online and productive fast, reduce costs, and maximize return on investment. IBM identity management provides identity life-cycle management (user self-care, enrollment, and provisioning), identity control (access and privacy control, single sign-on, and auditing), identity federation (sharing user authentication and attribute information between trusted Web services applications), and identity foundation (directory and workflow) to effectively manage internal users, as well as an increasing number of customers and partners through the Internet.

1.3.8 IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator synchronizes identity data residing in directories, databases, collaborative systems, applications used for human resources (HR), customer relationship management (CRM), and enterprise resource planning (ERP), and other corporate applications.

By serving as a flexible synchronization layer between a company's identity structure and the application sources of identity data, Tivoli Directory Integrator eliminates the need for a centralized datastore. For those enterprises that choose to deploy an enterprise directory solution, Tivoli Directory Integrator can help ease the process by connecting to the identity data from the various repositories throughout the organization.

1.4 Document structure

This redbook is divided into the following chapters:

▶ Chapter 1, "Portal security introduction" on page 1, this chapter, provides an overview of the scope of this book and products discussed in it.

▶ Chapter 2, "Portal security concepts" on page 13 explains some basic security concepts related to securing collaborative portal security.

▶ Chapter 3, "Implementation planning and considerations" on page 45 lists some issues that need to be addressed before implementing the security aspect of WebSphere Portal collaboration system and also describes the basic implementation options.

▶ Chapter 4, "Implementing and configuring basic LTPA authentication with IBM Directory Server" on page 57 explains some basic steps for implementing a collaborative portal with IBM Tivoli Directory Server.

▶ Chapter 5, "Setting up secure communication" on page 119 provides an overview of secure communication implementation with SSL over HTTP, LDAP, and DIIOP.

▶ Chapter 6, "Incorporating IBM Tivoli Access Manager for e-business" on page 167 introduces IBM Tivoli Access Manager to further secure a collaborative portal environment.

▶ Chapter 7, "Integrating directory servers in an IBM WebSphere Portal environment" on page 221 compares different implementations with different directory servers with the WebSphere Portal solution.

2

Portal security concepts

In this chapter, we discuss the issues regarding security in an IBM WebSphere Portal environment that also encompasses Lotus collaboration portlets. We divide this discussion into the following sections:

► Security concerns: Discusses the general security concerns in a WebSphere Portal environment.

► Communication encryption: Explains about communication encryption, certificates, and Secure Sockets Layer protocol.

► User identity and authentication: Describes directories that typically are used for authentication and considerations for a single sign-on process.

► Authorization topics: Provides some concepts about Java 2 Platform, Enterprise Edition security and authorization and how WebSphere Portal uses those facilities.

► Security facilities in portlets: Explores some programming facilities that relate to security and are available in the portlet context.

13

2.1 Security concerns

In the interconnected e-business environment, there are several basic security concerns that needs to be addressed. These concerns highlight the necessity for the security of Web applications, because without the ability to address them, the integrity of the applications themselves is compromised.

These concerns are:

► The ability to secure the communication traffic. Information flows into and from the application servers. The flow contains important and sensitive information, so intercepting this traffic might yield some security information, such as user ID information and account and identity information. Eavesdropping on traffic between servers is prevalent in the wired, networked world already and increases in the current wireless (Wi-Fi) world. The solution to secure communication is encryption and use of certificates. See 2.2, "Communication encryption" on page 15.

► The ability to authenticate the users accessing the application server. This concern has a two-sided effect: The lack of this function compromises the integrity of the application, while too much authentication hinders the end-user experience in using the application server. It is required that users are identified, and users must be identified once during their session to a set of application servers. This concern relates to user authentication and propagation, which is also known as single sign-on (SSO). See 2.3, "User identity and authentication" on page 24.

► The ability to differentiate user access and facilities. Each user might be allowed different functions and allowed access to different parts of the server. This concern is also enhanced in a portal environment, because each user can personalize his or her interface and might have different applications loaded. The application server needs to be able to extract the user authentication information and then collect access authorization information for the users. Typically, this is achieved using access control lists (ACLs) based on roles, users, or groups of users. See 2.4, "Authorization topics" on page 36.

There are additional security concerns that are important to protect a networked environment, but the implementation of these is typically independent of the application environment that is being protected. Due to its independent nature, we do not discuss the implementation of these additional security concerns in this book. These additional concerns include:

► The ability to control the communication traffic flow. This is typically performed by protocol filtering using firewalls. Firewalls allow selective network connections to be performed. The filtering can be generic, such as by port or protocol, or both, or it can be specific tunneling between hosts.

► The ability to quickly detect whenever a malicious attempt or attack is happening. This is typically called an intrusion detection system. The intrusion can be detected by firewalls, routers, application servers, or other components of the network. This information can be correlated and analyzed to bring up a comprehensive view of the network status.

We discuss the security concepts of these concerns in the following sections. Then, we discuss the specific collaborative portal interaction and the security concerns involved with this interaction.

2.2 Communication encryption

On unsecured networks such as TCP/IP, both the sender and the receiver need to be concerned about the security of the data that is sent over the network. The network protocol itself does not provide any protection against tampering with the data. In order to transport sensitive data over the network, the following issues need to be resolved:

► Confidentiality: Only the parties involved in the data transfer should be able to read the contents of the data.

► Authentication: The parties should be able to identify each other beyond doubt.

► Data integrity: Data that has been altered during transmission should be detectable.

► Non-repudiation: The sender of the data should not be able to deny the data he or she sends.

We first explore the theory of cryptography. We then discuss certificates and their relation to cryptography, and lastly, we describe the protocol that implements the cryptographic principles.

2.2.1 Cryptographic principles

Several cryptographic methods have been developed, and we describe some of these methods in the following sections.

Secret key cryptography
The oldest cryptographic technique is known as secret key cryptography. This is a way of achieving data confidentiality by encrypting and decrypting the data using a a single key (the symmetric or secret key). Both the sender and the receiver of the data must have the secret key. In Figure 2-1 on page 16, A is

sending a message to B, and both are using the same secret key to encrypt and decrypt the message.
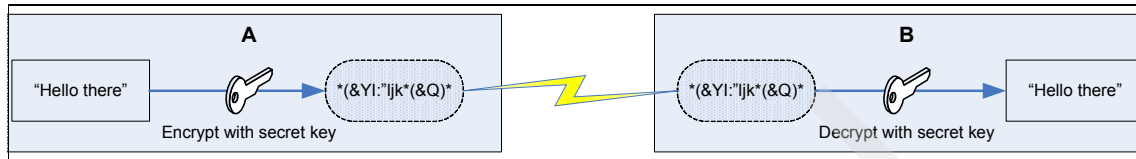
Figure 2-1   Secret key cryptography

No one else can encrypt or decrypt the message, because they do not have access to the secret key. The advantage of secret key cryptography is that it is fast. It is also less complex than public key cryptography, which is discussed in "Public key cryptography" on page 16.

The disadvantages of secret key cryptography are:

► The distribution of the secret keys. When one party creates a secret key, how is the other party going to get the secret key? Send it in e-mail? That is insecure. Send it with normal mail? That can take a long time and is hard to automate.

► Secret key administration. Another disadvantage is that for each party you want to communicate with, you must create a different secret key. If you do not, the different parties would be able to read the communications of all the parties. You will end up with a lot of keys, which can become cumbersome to manage.

Some well-known secret key algorithms are DES, Triple DES, Blow Fish, IDEA, and RC5.

Public key cryptography

Public key cryptography is another widely used cryptographic technique that does not have the problems of multiple shared secret keys and the distribution of secret keys. The most well-known public key algorithm is RSA, but a technique called Elliptic Curves is becoming more widespread.

Public key cryptography is a way of encrypting data using an asymmetric key pair. One key, the *public* key, is used for encrypting the data; the other key, the *private* key, is used for decrypting the data. Both keys are different, and the key used for encrypting data cannot be used for decrypting the same data. The public key is made available to the world, while the private key should never be revealed.

This technique is used for:

▶ Data encryption to provide confidentiality

As an example, in Figure 2-2, A is sending a message to B. The message is encrypted with B's public key, so only B can decrypt the message using B's private key.
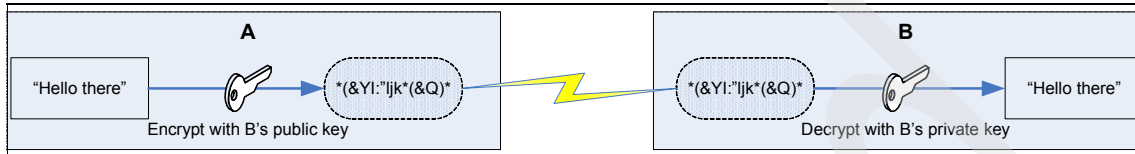
Figure 2-2   Public key cryptography

▶ Data signing for authentication of the sender

A, as the sender, wants to send data to B. A encrypts data with A's private key, and then B can decrypt the data with A's public key. This way, B is sure that the data came from A, because no one else can create a message that can be decrypted with A's public key. Figure 2-3 shows this process.

Note that although signing encrypts the data, it does not provide confidentiality, because anybody can decrypt the data by decrypting it with the sender's public key. What signing provides is non-repudiation and data integrity (the sender cannot deny sending the message). The message cannot be changed, because that would make it impossible to decode it.
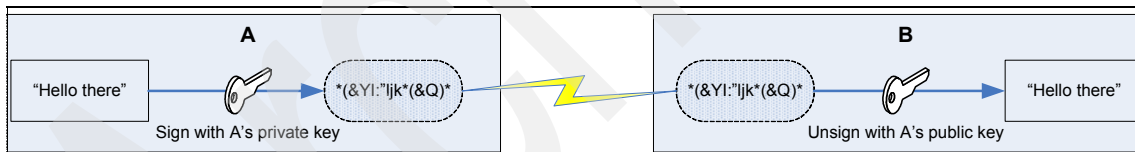
Figure 2-3   Signing data with public key cryptography

The combination of both processes can achieve confidentiality and non-repudiation. As shown in Figure 2-4 on page 18, when A sends a message to B, the encryption and decryption process is as follows:

1. A signs the data with A's private key.

2. A encrypts it with B's public key.

3. When B receives the signed and encrypted data, B decrypts it using B's private key.

4. B checks the signature by unsigning (decrypting) it with A's public key.
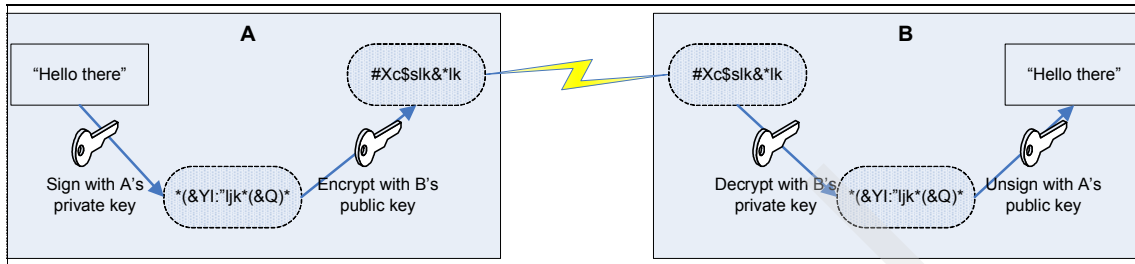
Figure 2-4   Signing and encryption using public key cryptography

Public key cryptography has some disadvantages, including:

▶ The public key cryptography process is computation intensive. Signing and encrypting the whole data stream implies a long encryption time. The difference in speed between secret key cryptography and public key cryptography is more than a factor of 1000.

▶ Man-in-the-middle attacks. Who guarantees that the public key of Alice is really the public key of Alice? Somebody might publish a public key that is claimed to be someone else's public key and access communications encrypted with that fraudulent public key.

Using hashes in digital signatures

We saw earlier that by encrypting a message with the private key of the sender, the message was signed, but also that this was a slow process due to the inherent slowness of public key cryptography. One way to overcome this problem is by creating a *message digest* through a hash function and subsequently sign the message digest.

A hash function is a tool that takes a message of any size and generates a small fixed-sized block of data (a message digest). A message digest has the following characteristics:

▶ A message digest is always the same for the same block of data.

▶ If the original message is altered, even by one bit, the resulting digest of the changed message is very different.

▶ Message digest creation through a hash function is very fast.

▶ It is impossible to generate the original message from the digest.

Therefore, instead of signing the whole message and encrypting it afterward (employing a digital signature), the encryption and decryption process shown in Figure 2-5 uses the following steps:

1. The sender, A, creates a message digest. A encrypts the digest with A's private key to create a digital signature.

2. The digital signature is appended to the original message, and the whole message is encrypted with B's public key.

3. The receiver, B, decrypts the whole message with B's private key and separates the digital signature from the message.

4. B decrypts the digital signature with A's public key to obtain the message digest. B also generates a message digest from the whole message with the same hash function.

5. B compares the two message digests. If both digests are the same, B is sure who the message comes from and that the message is unchanged.
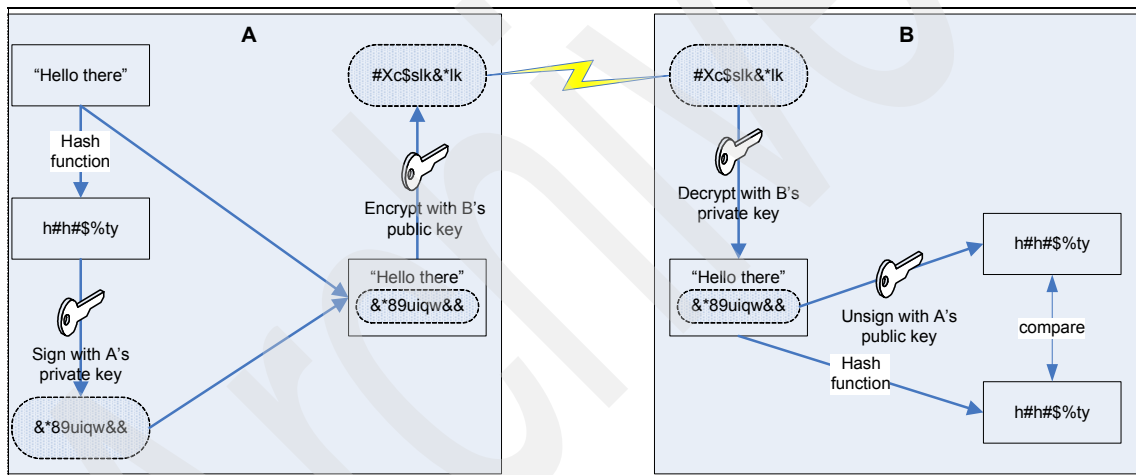
Figure 2-5   Public key cryptography using digital signatures

Some well-known hash algorithms are SHA-1, MD5, and RIPEMD.

Digital certificates

The problem of the authentication of the public key can be solved by the use of *digital certificates*. A digital certificate binds the owner of the public key to the public key itself. It is a data structure that contains a public key, necessary details about its owner, and some other information. All this information is signed by a

trusted third party, called a *certificate authority* (CA). Some important details about CAs include:

► When a public/private key pair is generated, the public key, together with the identity of the owner, must be submitted to a CA.

► The CA signs the data with the CA's own private key. The data becomes a digital certificate, and the CA returns it to the owner.

► A certificate does not contain any confidential data and should be made available to the world so that other people can use this certificate for sending data to the owner of the certificate and decrypting data from the owner.

There are many commercial CAs. There are also some local CAs for each country. Commercial CA certificates are often included in products such as Web browsers.

Adding secret key cryptography to the mix

We now have established a secure way to transfer messages through public key cryptography without having the authentication problems associated with it. The problem that remains is performance. It would be nice to marry the performance of secret key cryptography with the management scalability of public key cryptography. Here is how it works:

1. When two entities want to communicate, they set up communication channels using public key cryptography.

2. During this setup, a secret key is created and transferred to the parties involved. This secret key will only be used for this session and will be destroyed afterward.

3. After all parties have the secret key, they can send information to each other with secret key cryptography.

4. When the communication channel is closed, the secret key is destroyed.

2.2.2 Secure Sockets Layer protocol

In this section, we take a closer look about how the Secure Sockets Layer (SSL) protocol works. This protocol is used by the IBM Tivoli Web Services Manager components to communicate securely over untrusted networks.

SSL was conceived by Netscape Communications. It became the de facto standard to authenticate and encrypt communication between clients and servers on TCP/IP networks.

SSL performs the following functions:

► Authenticates the server to the client.

- ► Optionally, authenticates the client to the server.
- ► Creates an encrypted connection between both machines.

The authentication of the server to the client and vice versa happens through the exchange of certificates. The certificate authority that signed the certificate can be a different CA for the server than for the client. They must be trusted by the client and the server, respectively.

The encryption of the connection enables you to keep the data sent over the connection confidential. In addition, it also checks whether the data has been changed during the transfer.

All of these functions of SSL depend strongly on the cryptographic principles described in 2.2.1, "Cryptographic principles" on page 15.

SSL sits between the TCP/IP protocols, which are responsible for the transport and routing of data over the Internet, and the application protocols, such as Hypertext Transfer Protocol (HTTP). This is shown in Figure 2-6. The SSL structure in Figure 2-6 consists of two protocol levels:

- ► Record-layer protocol
- ► Communication protocols:
    - – Handshake protocol
    - – Change cipher specification protocol
    - – Alert protocol
    - – Application protocol
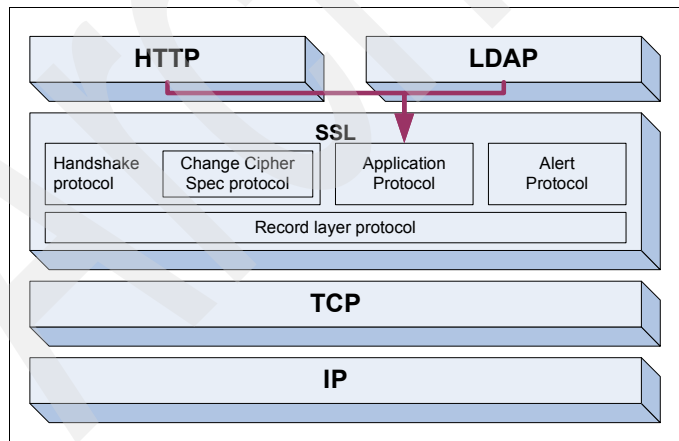
Figure 2-6   SSL structure and placement in the protocol stack

The record layer

All messages coming from the higher-level protocols go through the record layer before going to the transport layer. The record layer sends blocks of data called records, which are of fixed length. A record contains the content type, the protocol version number, the length, and the data, which is compressed and encrypted. Each message passes the following three functions:

► Fragmentation of data: The message is divided or combined to fit into a record. Remember that a record has a fixed length.
► Compression before sending the data.
► Encryption of the data part of the record.

The communication protocols

There are four communication protocols:

► The handshake protocol defines the sequence of events to establish an SSL session between two entities.

► The change cipher specification protocol is actually a subset of the handshake protocol. Its primary function is to indicate to the other party that there has been a change in the cryptographic options.

► The alert protocol deals with errors. An alert message contains two parts: the actual error description and the severity level of the error. There are two levels of errors:

– Warning: This indicates a potential problem. An example is the close_notify error, which specifies that the sender will not send any more messages in the current session.
– Fatal: This interrupts the current session and also means that the current session cannot be resumed in the future. An example of this is the bad_record_mac error, which indicates that the message or its hash has been tampered with.

► The application protocol is responsible for passing messages from the application-layer protocol to the record-layer protocol.

SSL communication is set up using the handshake protocol. Both entities negotiate the version of the protocol to be used (2.0 or 3.0), the cryptographic algorithms, and the setup of the keys. It is also possible to include entity authentication in this step (one-way or mutual). If this happens, the server will authenticate itself to the client using public key cryptography after which secret key cryptography is used for performance, as discussed in "Adding secret key cryptography to the mix" on page 20. The sequence of messages exchanged during the handshaking is illustrated in Figure 2-7 on page 23.
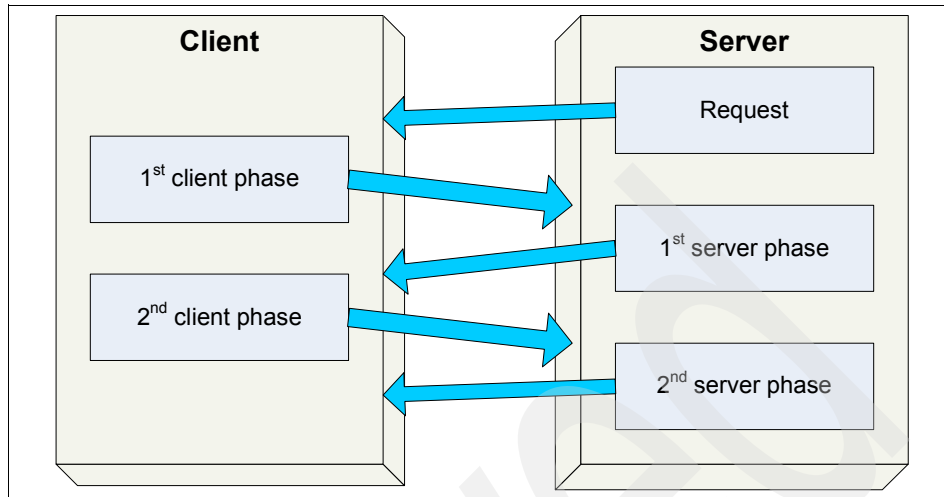
Figure 2-7   SSL 3.0 handshake protocol

The following list describes the sequence of messages:

1. A request is sent by the server to start the handshake process. This implies that the client has instigated the connection by requesting the appropriate URL from the server.

2. The first client phase reply contains:

   – A protocol version

   – A random number, used for the generation of the session keys

   – A session ID, to check wether a new session needs to be set up

   – A list of cryptographic options: key exchange algorithm, hash algorithm, and so on

   – A list of the compression methods supported by the client

   After sending this message, the client waits for the first server phase. If it receives any other type of message, this results in a fatal error, and the handshake must be restarted.

3. During the first server phase, the server sends the following messages:

   a. It acknowledges which version of the SSL protocol is supported (lower than or equal to the version of the client). The server also generates a random number and returns the session ID from the first client phase, if it agrees to pick up the old session. A choice is also made from the set of cryptographic options and compression methods that were proposed by the client.

b. If the server authenticates to the client, the certificate of its public key is included.

c. With the server key exchange, the key exchange algorithm is specified. It can either be Diffie-Hellmann, RSA, or Fortezza.

d. If the server wants the client to authenticate (mutual authentication), it sends a certificate request.

4. The client responds with the second client phase:

a. If the server requested it, the client sends its certificate.

b. With the certificate verify, the client proves to the server that it is in possession of the private key that corresponds to the public key in its certificate by digitally signing a specific message (called a challenge). See "Digital certificates" on page 19 for more details about digital certificates.

c. The client key exchange contains all the necessary information that both parties need to calculate all the session keys (the ephemeral secret keys). See "Adding secret key cryptography to the mix" on page 20 for more details about this step.

5. Both parties now send, in turn, the finished message, preceded by the change cipher spec message, as shown in Figure 2-7 on page 23. These are the first messages that use the newly agreed upon key material.

SSL is a session-based protocol and not a connection-based protocol. A connection is set up every time there is data to be transferred between the client and the server. It is not necessary to go through the handshake protocol every time a new connection is made. Therefore, if previously arranged keys and algorithms are used, the session is resumed.

2.3 User identity and authentication

The user authentication involves checking the users' identities and ensuring that this information is propagated to any application with the need to know. We divide the discussion in this section into the following topics:

► Directories
► WebSphere Member Manager
► Credential Vault mechanism

2.3.1 Directories

Information describing the various users, applications, files, printers, and other resources accessible from a network is often collected into a special database

that is sometimes called a directory. As the number of different networks and applications has grown, the number of specialized directories of information has also grown. This growth results in islands of information that are difficult to share and manage. If all of this information could be maintained and accessed in a consistent and controlled manner, it would provide a focal point for integrating a distributed environment into a consistent and seamless system.

Introducing LDAP

Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet these needs. LDAP defines a standard method for accessing and updating information in a directory. LDAP is gaining wide acceptance as the directory access method of the Internet and is, therefore, also becoming strategic within corporate intranets. This method is supported by a growing number of software vendors and is being increasingly incorporated into applications.

LDAP defines a message protocol used by directory clients and directory servers. LDAP uses a variety of messages. For example, a bindRequest might be sent from the client to the LDAP server at the beginning of a connection. A searchRequest is used to search for a specific entry in the directory.

There are also associated LDAP APIs for the C language and ways to access LDAP from within a Java application. In addition, within the Microsoft development environment, you can access LDAP directories through its Active Directory Service Interface (ADSI). In general, with LDAP, the client is not dependent on a particular implementation of the server; the server can implement the directory however it chooses.

LDAP defines a communication protocol. That is, it defines the transport and format of messages used by a client to access data in an X.500-like directory. LDAP does not define the directory service itself. When people talk about the LDAP directory, they refer to the information that is stored and can be retrieved by the LDAP protocol.

All modern LDAP directory servers are based on LDAP Version 3. You can use a Version 2 client with a Version 3 server. However, you cannot use a Version 3 client with a Version 2 server unless you bind as a Version 2 client and use only Version 2 APIs.

All LDAP servers share many basic characteristics because they are based on industry-standard Request for Comments (RFC). However, due to implementation differences, they are not all completely compatible with each other when there is not a standard defined.

Common directory benefits

If application developers could be assured of the existence of a directory service, application-specific directories would not be necessary. However, a common directory must address the previously mentioned problems. It must be based on an open standard that is supported by many vendors on many platforms. The common directory must be accessible through a standard API. It must be extensible so that it can hold the types of data needed by arbitrary applications. Also, this directory must provide full functionality without requiring excessive resources on smaller systems. Because more users and applications will access and depend on the common directory, it must also be robust, secure, and scalable.

When such a directory infrastructure is in place, application developers can devote their time to developing applications instead of application-specific directories. In the same way that developers rely on the communications infrastructure of TCP/IP and remote procedure call (RPC) to free them from low-level communication issues, they must be able to rely on powerful, full-function directory services. LDAP is the protocol to be used to access this common directory infrastructure. Like HTTP and File Transfer Protocol (FTP), LDAP has become an indispensable part of the Internet's protocol suite.

When applications access a standard common directory that is designed in a proper way instead of using application-specific directories, redundant and costly administration can be eliminated, and security risks are more controllable. For example, the telephone directory, mail, and Web applications shown in Figure 2-8 on page 27 can all access the same directory to retrieve an e-mail address or other information stored in a single directory entry. The advantage is that the data is kept and maintained in one place. Various applications can use individual attributes of an entry for different purposes (assuming that the they have the correct authority). New uses for directory information will be realized, and a synergy will develop as more applications take advantage of the common directory. Figure 2-8 on page 27 depicts the advantages of this arrangement.
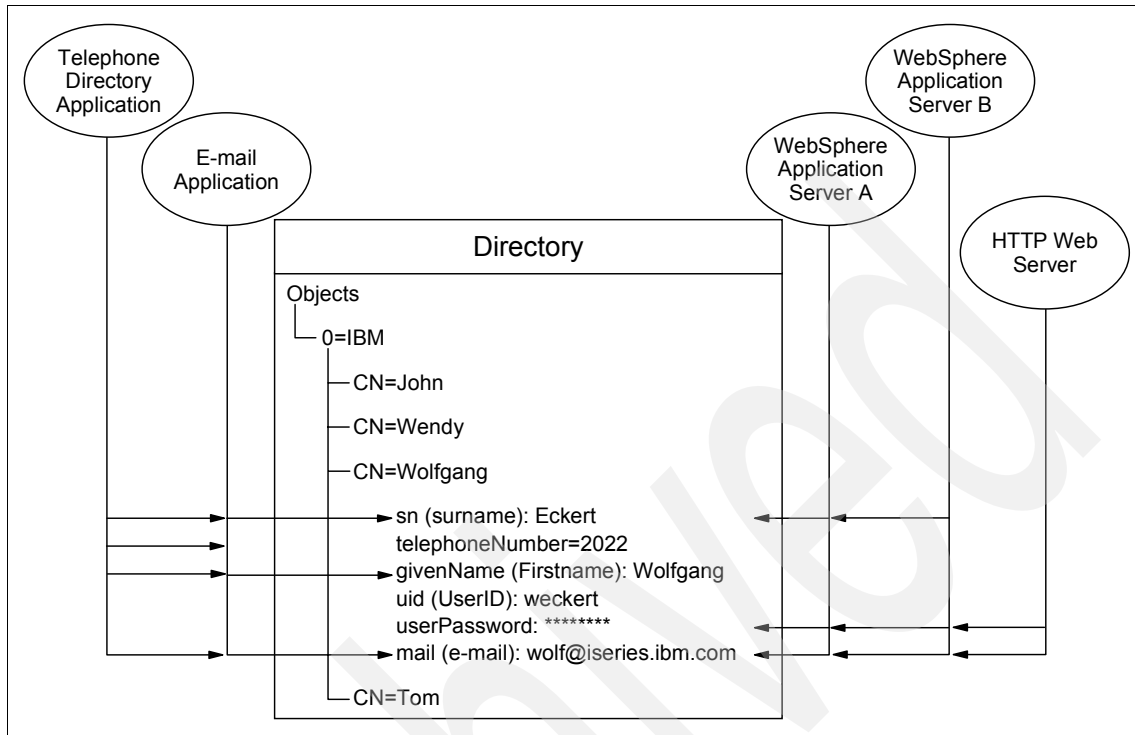
Figure 2-8   Several applications using attributes of the same entry

Storing data in a directory and sharing it among applications saves you time and money by keeping administration effort and system resources down. Many IBM applications also utilize directories to centrally store and share information. The number of applications that support LDAP directories is constantly increasing. For example, LDAP directory support, such as for authentication and configuration management, is provided in various IBM operating systems, IBM WebSphere Application Server, WebSphere Portal, Tivoli Access Manager, Directory Server, IBM HTTP Server, Lotus Domino, and so on.

LDAP directory structure

A directory contains a collection of objects organized in a tree structure. The LDAP naming model defines how entries are identified and organized. Entries are organized in a tree-like structure called the Directory Information Tree (DIT). Entries are arranged within the DIT based on their distinguished name (DN). A DN is a unique name that unambiguously identifies a single entry. DNs are made up of a sequence of relative distinguished names (RDNs). Each RDN in a DN corresponds to a branch in the DIT leading from the root of the DIT to the

directory entry. A DN is composed of a sequence of RDNs separated by commas, such as `cn=thomas,ou=itso,o=ibm`.

You can identify common names (CNs) within DNs. You also can organize entries, for example, after organizations. You can further split the tree into organizational units within a single organization as needed. You can define your DIT based on your organizational needs as in the simple example shown in Figure 2-9. If you have, for example, one company with different divisions, you might want to start with your company name under the root as the organization (o) and then branch into organizational units (ou) for the individual divisions. In case you store data for multiple organizations within a country, you might want to start with a country (c) and then branch into organizations. Figure 2-9 provides an example of this approach.
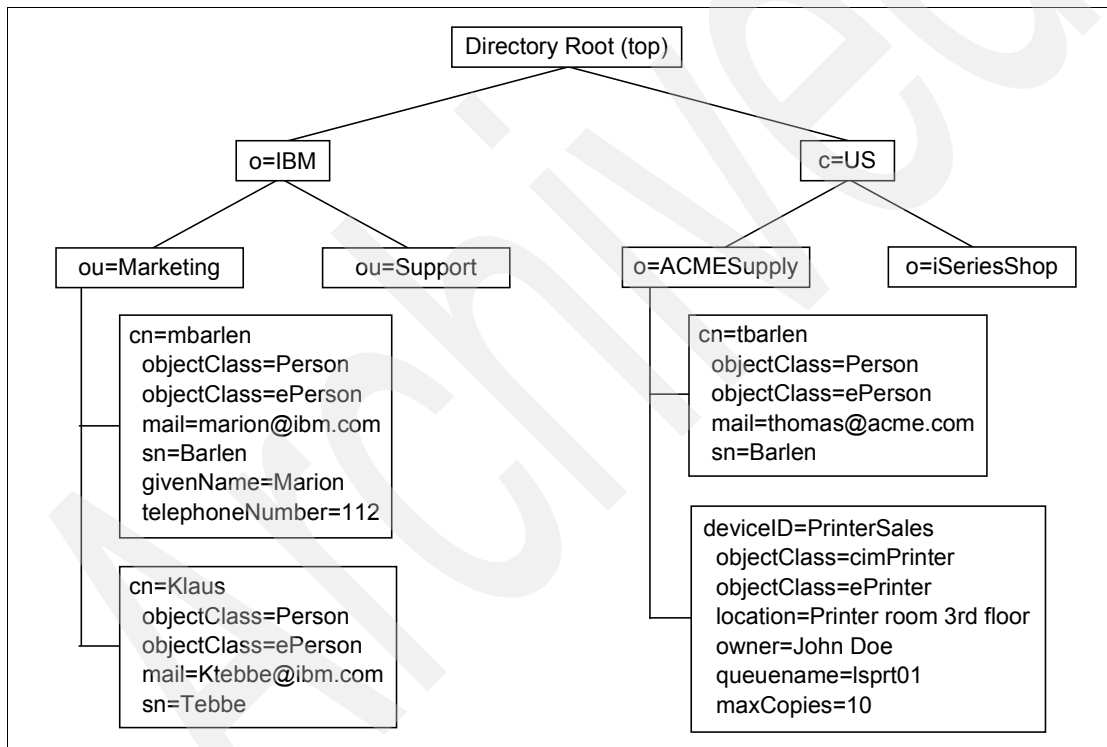
Figure 2-9   Example of a directory information tree

Each object, also referred to as an entry in a directory, belongs to one or more object classes. An object class describes the content and purpose of the object. It also contains a list of attributes, such as a telephone number or surname, that can be defined in an object of that class. You can publish entries of different object classes under another object.

The object class also defines which of the attributes must be defined (are required) when creating an object of this class and which attributes are optional. Object classes also can inherit characteristics, such as attributes from other object classes. In the example of the ePrinter, the class inherits all of the attributes that are defined in the class cimPrinter. Therefore, you must define the deviceID when you create an ePrinter object. Optionally, you can specify the location, owner, and queuePtr attribute of ePerson and all of the attributes of cimPrinter.

Attributes themselves also have certain characteristics. The surname attribute name, for example, is defined as sn and surName and describes a person's family name. The attribute definition also specifies the syntax rules for the attribute value. A telephone number can only contain numbers and hyphens, while the surname consists of alphabetic characters. Other specifications include whether this attribute can contain only one or many values, the matching rules, the object identifier (OID), and so on. The IBM Tivoli Directory Server product also includes some IBM proprietary extensions for each attribute. Other manufacturers, such as Microsoft, have similar extensions. The IBM extensions also include an access class that is used in combination with access control lists (ACLs) to control who can perform a certain action on the attribute value (such as read, write, search, or compare operations).

All the objects and attributes with their characteristics are defined in schemas. The schema specifies what can be stored in the directory. Schema-checking ensures that all required attributes for an entry are present before an entry is stored. Schema-checking also ensures that attributes not in the schema are not stored in the entry. Optional attributes can be filled in at any time. A schema also defines the following:

▶ Inheritance
▶ Subclassing of objects
▶ Where in the DIT structure (hierarchy) objects can appear

LDAP implementations

In this redbook, we discuss LDAP server implementation based on IBM Tivoli Directory Server, IBM Lotus Domino, and Microsoft Active Directory.

2.3.2 WebSphere Member Manager

WebSphere Member Manager is a standard feature of WebSphere Application Server that encapsulates access to directory servers. It uses a standard LDAP schema, inetOrgPerson, that is based on RFC 2798.

The implementation of WebSphere Member Manager uses a relational database for storing WebSphere-based attributes that are not stored in the LDAP directory.

By default, the information is stored in Cloudscape™ database. The DB2-based information is stored in the WPS50 database from WebSphere Portal.

The WebSphere Portal People Finder and People and Group management use WebSphere Member Manager.

2.3.3 Credential Vault mechanism

When integrating different back-end systems, portlets often need to provide some type of authentication to access these back-end systems. WebSphere Portal provides the use of a Credential Vault to store and retrieve user credentials. By using Credential Vault portlets, you can provide a single sign-on experience to the user.

This section describes:

► The value of the Credential Vault for portlet development

► Components of the Credential Vault

► Credential Vault objects

Portlets running on WebSphere Portal might need to access remote applications that require some form of authentication by using appropriate credentials. In this section, we provide an overview of the Credential Vault components.

Credentials

Examples of credentials are user IDs and passwords, SSL client certificates, and private keys. In order to provide a single sign-on user experience, portlets should not ask the user for the credentials of individual applications each time the user starts a new portal session. Instead, they must be able to store and retrieve user credentials for their particular associated application and use those credentials to log in on behalf of the user. The portal back-end secure access is illustrated in Figure 2-10 on page 31.
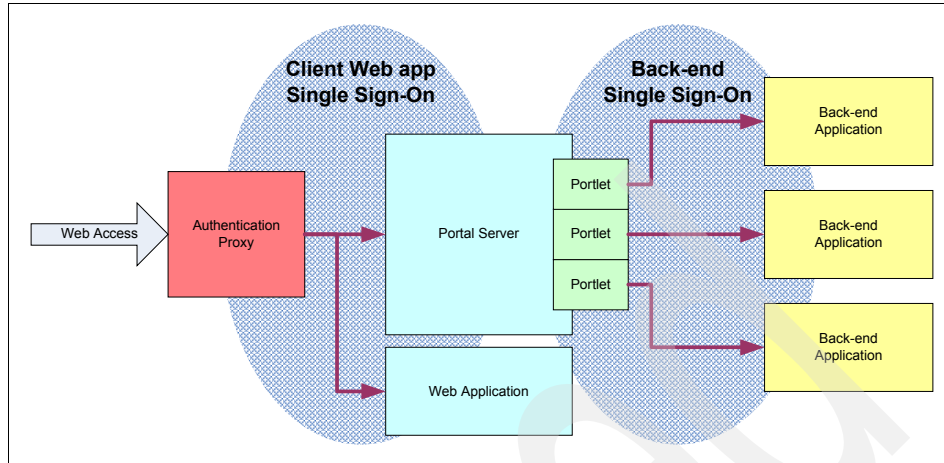
Figure 2-10   Credential Vault in action

The Credential Vault provides this functionality, and portlets can use it through the Credential Vault Portlet Service.

Components of the Credential Vault organization

The organization of the Credential Vault in WebSphere Portal consists of vault segments and credential slots. Figure 2-11 on page 32 shows an overview of these components.
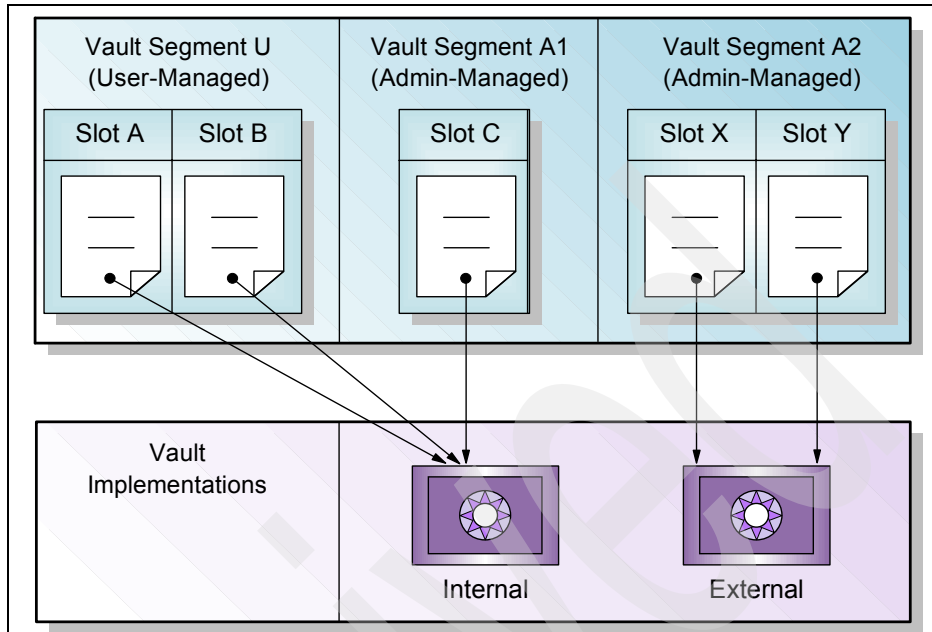
Figure 2-11   Credential Vault organization

As shown in Figure 2-11, the components of a Credential Vault are:

► Vault segments: The Credential Vault is partitioned into segments, and a vault segment contains one or more credential slots. There are two different types of vault segments:

   – Administrator-managed: The creation of new slots is restricted to the portlet administrator.

   – User-managed: Portlets can also create new slots on behalf of the user.

Note: Setting and retrieving credentials can be performed by portlets for both types of vault segments.

Vault implementations are the actual locations where the credentials are stored. This can be, for example, the default database of WebSphere Portal or the Tivoli Access Manager lock box.

- ► Credential slots: As mentioned previously, every vault segment contains one or more credential slots. Slots are "drawers" where portlets store and retrieve a user's credentials. Each slot holds one credential and links to a resource in a vault implementation. There are four different types of slots:
  - A system slot stores system credentials where the actual secret is shared among all users and portlets.
  - An administrative slot allows each user to store a secret for an administrator-defined resource (for example, Lotus Notes®).
  - A shared slot stores user credentials that are shared among the user's portlets.
  - A portlet private slot stores user credentials that are not shared among portlets.
- ► Credentials objects: WebSphere Portal differentiates between passive and active credential objects:
  - *Passive credential objects* are containers for the credential's secret. Portlets that use passive credentials need to extract the secret out of the credential and do all the authentication communication with the back-end resource. The following passive credential support is provided with WebSphere Portal:
    - UserPasswordPassive, which stores secrets in the form of user ID/password pairs
    - SimplePassive, which stores secrets in the form of serializable Java objects
    - JaasSubjectPassive (Java Authentication and Authorization Service), which stores secrets in form of javax.security.auth.Subject objects

    Currently, the vault service in WebSphere Portal only supports UserPasswordPassive.
  - *Active credential objects* hide the credential's secret from the portlet; there is no way of extracting it out of the credential. In return, active credential objects offer business methods that take care of all the authentication. The following active credential support is provided with WebSphere Portal:
    - HttpBasicAuth
    - HttpFormBasedAuth
    - JavaMail
    - LtpaToken
    - SiteMinderToken
    - WebSealToken

    When using active credentials, portlets never get in touch with the credential secrets, and thus, there is no risk a portlet could violate any security rules, such as storing the secret on the portlet session. Although

there might not always be an appropriate active credential class available, this is the preferred type of credential objects to use.

2.3.4 Lightweight Third Party Authentication token

The mechanism used by WebSphere Portal to work with Lotus Collaborative application are primarily uses the Lightweight Third Party Authentication (LTPA) token.

As the name indicates, the authentication is performed by one of the components, and that component passed the authenticated user information for the back-end processes. The LTPA token is the package that contains this information. It is implemented in the form of a session cookie that passes through the browser that is included in the page request. Figure 2-12 show this mechanism.
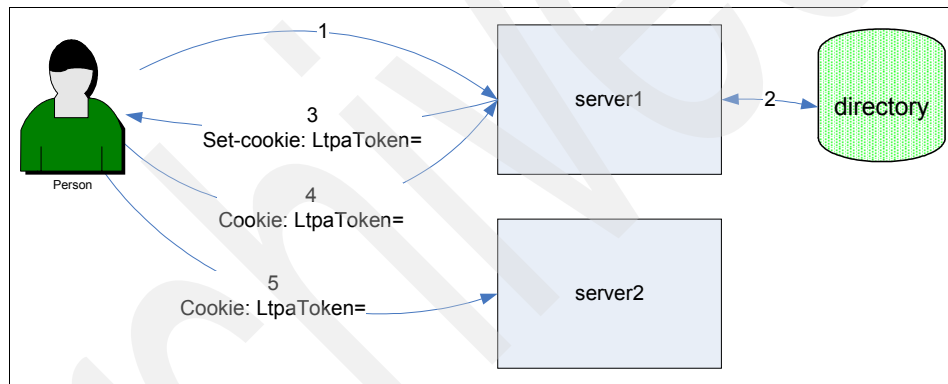
Figure 2-12   LTPA token passing

As shown in Figure 2-12, the following steps describe the mechanism:

1. An end user, using a Web browser, enters the user ID and password to be authenticated by server1.

2. Server1 authenticates the user ID and password to an LDAP directory server.

3. When the user ID and password is verified, server1 creates an LTPA token that is sent back to the Web browser as a session cookie. This is accomplished in the HTTP header Set-Cookie directive.

4. The Web-browser that receives the Set-Cookie directive sends the LTPA token cookie in the all subsequent conversations to server1. This way, server1 can decode the user identity.

5. When the user requires access to server2, the cookie will be passed to server2. Because server1 and server2 already agreed on the encryption key, server2 can decode the passed LTPA token and authenticate the user. Server2 can then verify that the user record exists in the directory and accept the user.

2.3.5 Trust Association Interceptor

The use of the Trust Association Interceptor (TAI) is similar to LTPA token passing. However, the credential is passed in the HTTP header's basic authentication field, instead of being passed as a cookie.

If authentication is being done on a separate system, the server does not need to do the exact same authentication as long as it knows requests come the authentication system. The server authentication process should be modified to execute two tasks:

► Make sure that requests come from the authentication server

► Get the identity of users from requests

This process is called the Trust Association Interceptor (TAI). The TAI feature is, in essence, a point in the authentication process where an organization can insert their own code to achieve whatever authentication outcome they desire. As the name implies, WebSphere is going to *trust* the result returned to it by the TAI. Therefore, WebSphere needs to trust the server that did the authentication.

We recommend that you establish a level of trust by enforcing a trusted connection between these two parties. This connection could, for example, be a VPN or specially secured network segments.

TAI is coded as a Java class. In a WebSphere environment, there can be several TAI classes active. Each TAI class has these three main Java methods:

► isTargetInterceptor
This method determines whether this TAI class is to process the request received. If it returns a value of false, WebSphere invokes the next Trust Association Interceptor, if any are specified, or processing returns to WebSphere, which then performs standard authentication processing. This method can use the HTTPServletRequest object, which contains the HTTP headers of the request. The method can use information obtained from this object to make its decision as to whether or not it will process the request.

- ► validateEstablishedTrust
  Having decided that this class is to process the request, this method determines if the request is valid. Rather than having WebSphere perform its own authentication, WebSphere relies on this method to verify that the request it is processing comes from a trusted source, where authentication was successful. This method can also use the HTTPServletRequest object. How it does this depends on how authentication has been done by the remote authenticating process. Typically, the remote authentication process will need to add some HTTP header tag to the request after it completes authentication. This method would then know to look for that HTTP header tag and validate it.

- ► getAuthenticatedUserName
  Having decided that the request is valid, the purpose of this method is to return a user ID that will then be used by WebSphere. The user ID needs to be a valid user ID in the security system.

2.4 Authorization topics

Authorization in WebSphere Application Server is based on the Java 2 Platform, Enterprise Edition (J2EE) security infrastructure. We discuss authorization in the WebSphere environment for the following topics:

- ► Java 2 Platform, Enterprise Edition security
- ► IBM Tivoli Access Manager for e-business

2.4.1 Java 2 Platform, Enterprise Edition security

The Java 2 Platform, Enterprise Edition (J2EE) specification defines the building blocks and elements of a J2EE application that build an enterprise application. The specification also provides details about security related to the different elements.

The J2EE application consists of multiple modules and components; these elements are in connection with each other, and they communicate through certain protocols. This section only discusses the connection on the application level, without going into details about the protocols.

For example, a user accesses a JSP on the application server. This JSP is a secured resource. In this situation, the application server has to authenticate the user and decide whether the user is authorized to access the page or not. In this case, the connection between the user's browser and the JSP page requires security.

In another example, a servlet in the Web container on the application server accesses an EJB in the EJB container on the application server. The same thing happens as in the previous example; the application server has to authenticate the servlet's request on behalf of the EJB, and then check the authorization.

When you design an enterprise application or security for an application, you will have a similar, but more detailed diagram for your solution. Make sure that you have taken every connection into consideration between each element and module. Security in this context consists of two major parts: authentication and authorization. Make sure that the access is always authenticated or the security credentials are propagated. In addition, ensure that the access is authorized and prepare an action if authorization is not granted.

For more information, read the security related sections of the Java 2 Platform Specification V1.3 at:

    http://java.sun.com/j2ee/docs.html

Security roles

The J2EE Specification defines a security role as: "A logical groupings of users that are defined by an Application Component Provider or Assembler." Security roles provide a mechanism whereby application developers determine the security policies for an application by creating named sets of users (for example, managers, customers, and employees) that will have access to secure resources and methods. At application assembly time, these sets of users, or security roles, are not tied to any real users or groups of users. Instead, they are placeholders that are later mapped to real users and groups at application deployment time, during a process called *security role mapping*.
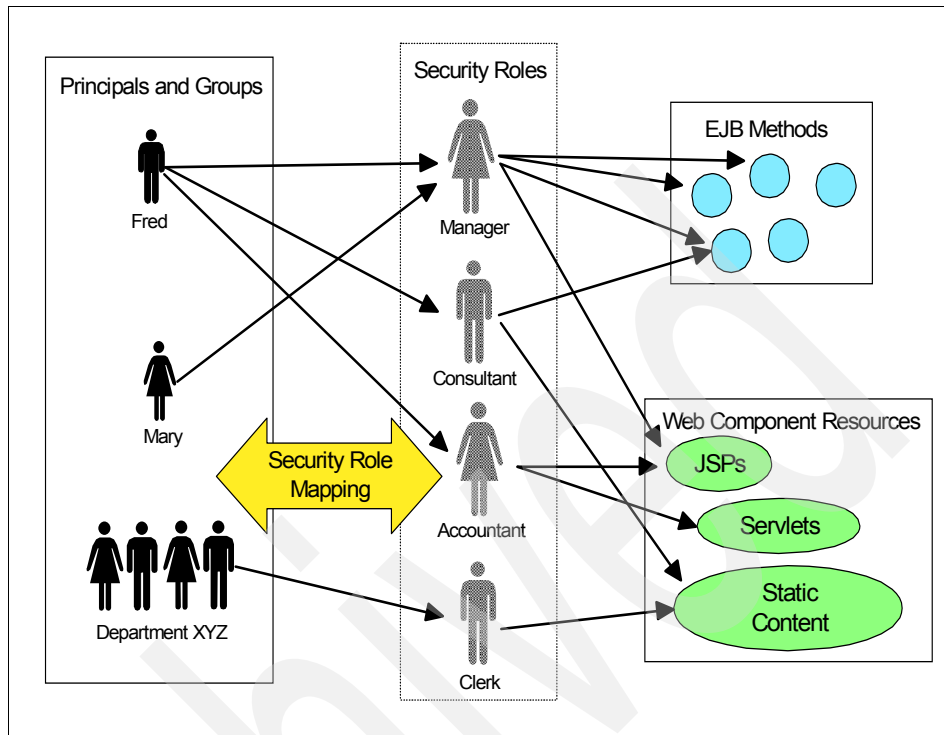
Figure 2-13   Security roles

This two-phase security administration approach allows for a great deal of flexibility and portability. Deployers of an application have full control over how their local users and groups are mapped to the application's security roles, and over what authorization and authentication mechanisms are used to determine role membership.

At deployment time, security roles can be mapped to users, groups of users, or *special subjects.* There are two special subjects in WebSphere Portal Version 5:

► All Authenticated Users
► Everyone

J2EE container-based security

J2EE containers are responsible for enforcing access control on component objects and methods. Containers provide two types of security:

► Declarative security

Declarative security is the means by which an application's security policies can be expressed externally to the application code. At application assembly time, security policies are defined in an application's *deployment descriptor*. A deployment descriptor is an XML file that includes a representation of an application's security requirements, including the application's security roles, access control, and authentication requirements.

When using declarative security, application developers are free to write component methods that are completely unaware of security. By making changes to the deployment descriptor, an application's security environment can be radically changed without requiring any changes in application code.

► Programmatic security

Programmatic security is used when an application must be "security-aware." For example, a method might need to know the identity of the caller for logging purposes, or it might perform additional actions based on the caller's role. The J2EE Specification provides an API that includes methods for determining both the caller's identity and the caller's role.

The EJB methods are:

– isCallerInRole
– getCallerPrincipal

The HttpServlet methods are:

– isUserInRole
– getUserPrincipal

Role association

There are two deployment descriptor files used for security role mapping, as shown in Table 2-1.

Table 2-1   Role mappings in deployment descriptors

File name	Purpose	Mandatory?
application.xml	Security roles defined	Yes.
ibm-application-bnd.xmi	Security roles mapped	No. Security roles can be mapped during or after installation.

In the application.xml file, all security roles used in the application must be named, with an optional description. Example 2-1 shows the XML elements required to define six security roles: manager, consultant, clerk, accountant, allauthenticated, and everyone.

Example 2-1   Security role definitions in the application.xml file

```
<security-role id="SecurityRole_1">
    <description>ITSOBank manager</description>
    <role-name>manager</role-name>
<security-role>
<security-role id="SecurityRole_2">
    <description>ITSOBank consultant</description>
    <role-name>consultant</role-name>
</security-role>
<security-role id="SecurityRole_3">
    <description>ITSOBank clerk</description>
    <role-name>clerk</role-name>
</security-role>
<security-role id="SecurityRole_4">
    <description>ITSOBank accountant</description>
    <role-name>accountant</role-name>
</security-role>
<security-role id="SecurityRole_5">
    <description>All authenticated users</description>
    <role-name>allauthenticated</role-name>
</security-role>
<security-role id="SecurityRole_6">
    <description></description>
    <role-name>everyone</role-name>
</security-role>
```

In the ibm-application-bnd.xmi file, security roles are mapped to users or groups in the user registry. Table 2-2 shows how the security roles defined previously would be mapped.

Table 2-2   Role mappings

Security role	Mapped to
manager	managergrp
consultant	consultantgrp
clerk	clerkgrp
accountant	accountantgrp
allauthenticated	All Authenticated Users (special subject)
everyone	Everyone (special subject)

Example 2-2 shows a code snippet from the ibm-application-bnd.xml file that
holds the binding information for the J2EE roles.

Example 2-2   Security role mappings in the ibm-application-bnd.xmi file

```
<authorizationTable xmi:id="AuthorizationTable_1">
    <authorizations xmi:id="RoleAssignment_1">
        <role href="META-INF/application.xml#SecurityRole_1"/>
        <groups xmi:id="Group_1" name="managergrp"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_2">
        <role href="META-INF/application.xml#SecurityRole_2"/>
        <groups xmi:id="Group_2" name="consultantgrp"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_3">
        <role href="META-INF/application.xml#SecurityRole_3"/>
        <groups xmi:id="Group_3" name="clerkgrp"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_4">
        <role href="META-INF/application.xml#SecurityRole_4"/>
        <groups xmi:id="Group_4" name="accountantgrp"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_5">
        <specialSubjects xmi:type="applicationbnd:AllAuthenticatedUsers"
xmi:id="AllAuthenticatedUsers_1" name="AllAuthenticatedUsers"/>
        <role href="META-INF/application.xml#SecurityRole_5"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_6">
        <specialSubjects xmi:type="applicationbnd:Everyone" xmi:id="Everyone_1"
name="Everyone"/>
        <role href="META-INF/application.xml#SecurityRole_6"/>
    </authorizations>
</authorizationTable>
```

2.4.2 IBM Tivoli Access Manager for e-business

IBM Tivoli Access Manager for e-business can act as the front-end authentication
and authorization agent for a Web server. It uses a reverse proxy that provides a
layer of security in the DMZ, while the actual Web servers and Web application
servers are put behind firewalls. The conceptual scenario of IBM Tivoli Access
Manager for e-business is shown in Figure 2-14 on page 42.
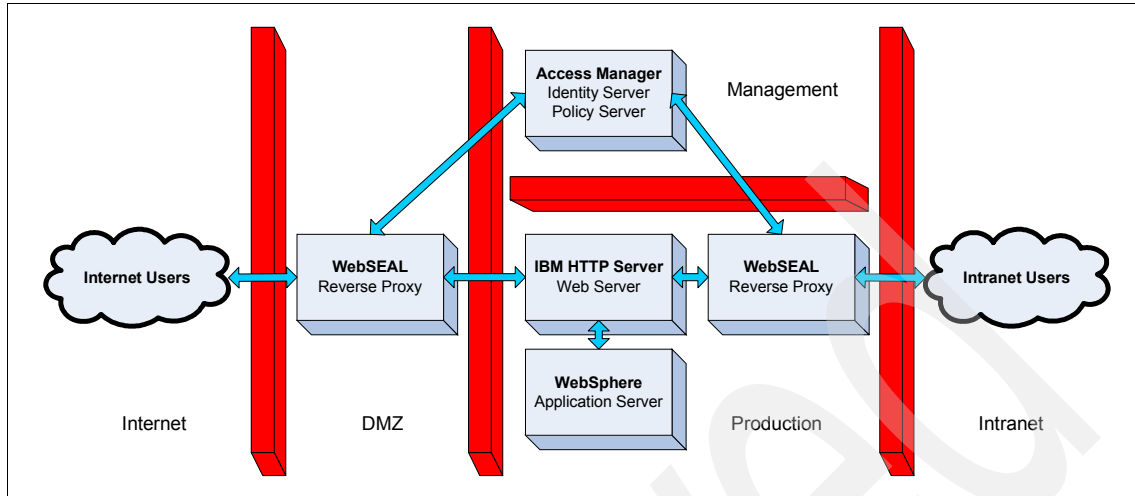
Figure 2-14   Conceptual diagram of IBM Tivoli Access Manager for e-business

In Figure 2-14, the network is divided, using firewalls, into several regions:

Internet   This is where external access originates.

Demilitarized zone   This is where external access is allowed, while security and access control is performed within a protocol firewall. This is typically the initial line of defense.

Production zone   This is where the critical application servers resides. Access to this area is restricted to machines in the DMZ or management zone only.

Management zone   This is where the authentication and authorization components should reside. No external access should be allowed, because this might compromise the security. Access to these machines should be performed physically or through a specialized VPN connection.

Intranet   This is for internal user access. Some companies allow intranet users to access the production zone directly, while others create a separate intranet DMZ.

The reverse proxy node, also called WebSEAL, is put in the DMZ to receive external access. It forwards requests based on specific junctions that are created and mapped to servers in other zones, typically the Web servers and Web application servers.

The firewalls then can be configured to allow specific traffic to pass from certain machine pairs to ensure authenticity and access.

2.5 Security facilities in portlets

As has been discussed in the previous sections, there are some basic security issues regarding a general Web application. Additional security measures can be taken in the portlets context, such as:

► WebSphere Member Manager facilities
► Credential Vault interface to access back-end systems

3

Implementation planning and considerations

This chapter describes some planning and other considerations about how to secure a collaborative portal implementation. You can choose several different implementation options in this area. We divide the discussion into the following topics:

► Planning

In this section, we discuss the implementation prerequisites and software level that we used in this redbook. We also provide additional information about path name usage and substitution.

► Collaborative portal interaction

► Implementation options

45

3.1 Planning

In our collaborative portal installation, the implementation is based on the configuration shown in Figure 3-1.
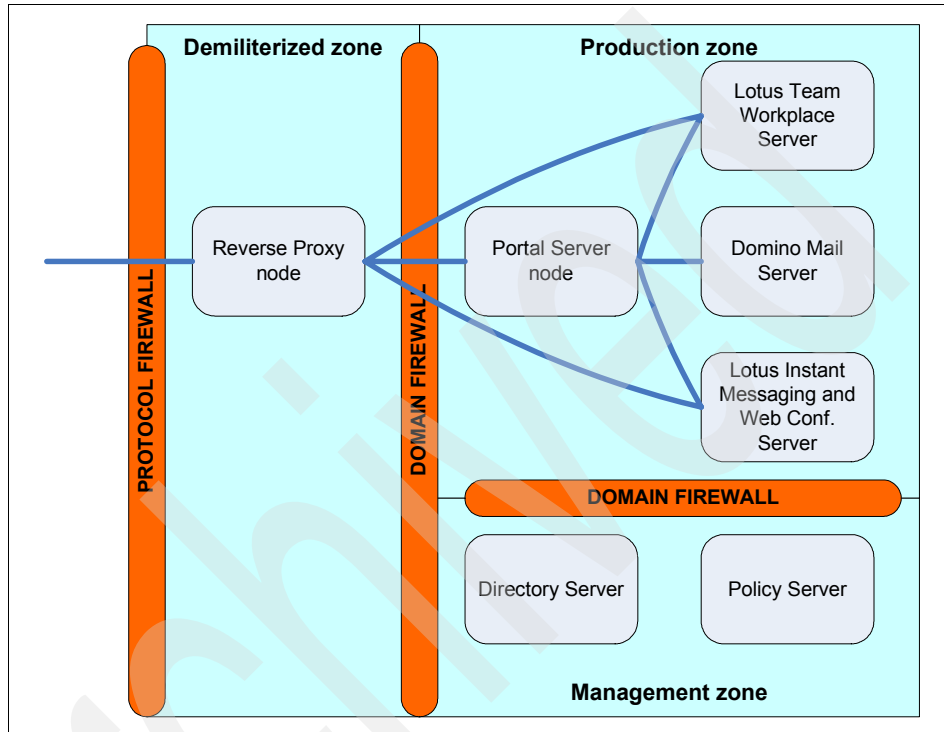
Figure 3-1   Implementation configuration

In Figure 3-1, the components that we used are:

► Reverse proxy node: This is where the authorization and authentication agents run, and this node acts as the gateway to the outside world.

► WebSphere Portal server node: This is where the WebSphere Portal application is installed. In our scenario, we only have one WebSphere Portal server node, and the Web server is all included in this machine. In the production environment, you might need to separate the Web server and have multiple WebSphere Portal server nodes.

► Domino mail server: This is the primary Domino server that is also serving the e-mail application for Domino.

► Lotus Team Workplace server: This is another Domino server that hosts the Lotus Team Workplace (formerly called QuickPlace) application.

- ► Lotus Instant Messaging and Web Conferencing server: This is another Domino server that hosts the Lotus Instant Messaging and Web Conferencing (formerly called Sametime) application.

- ► Directory server: This is where the user identities and registries are stored and can serve other component for authentication.

- ► Policy Server: This is where the external authorization server is located to provide access rights for the components.

> **Note:** We do not include procedures for implementing the firewalls.

3.1.1 Hardware and software prerequisites

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

- ► *IBM Tivoli Access Manager Base Installation Guide, V5.1*, SC32-1362

- ► *IBM Tivoli Access Manager for e-business Web Security Installation Guide, V5.1*, SC32-1361

- ► *IBM Tivoli Directory Server Installation and Configuration Guide, 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

- ► Lotus documentation library:

  http://www.lotus.com/ldd/doc

3.1.2 Software used in the our run-time environment

We implemented the collaborative portal environment using Microsoft Windows® 2000 Server with Service Pack 4 on all our servers with all the critical patches applied. The software installed in the individual machines includes:

- ► Policy server node:
  - – IBM DB2 UDB, Enterprise Server Edition Version 8.1.4.428 (Version 8.1 with Fix Pack 4a)
  - – IBM GSKit Version 7.0.1.16
  - – IBM Java Runtime Environment (JRE) Version 1.3.1
  - – IBM WebSphere Application Server Version 5.0.2 (Version 5.0 with Fix Pack 2 for the Web Administration Client)

– IBM Tivoli Directory Server Version 5.2 with the following features installed:

- Directory Server
- Directory Client SDK
- Web Administration Tool

– IBM Tivoli Access Manager for e-business Version 5.1.0.2 (Version 5.1 and Base Fix Pack 2) with the following features installed:

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

► Reverse proxy node:

– IBM GSKit Version 7.0.1.16

– IBM Java Runtime Environment (JRE) Version 1.3.1

– IBM Tivoli Access Manager for e-business Version 5.1.0.2 (Version 5.1 with WebSEAL Fix Pack 2 and Base Fix Pack 2) containing the following features:

- Access Manager Runtime
- Access Manager Java Runtime Environment (PDJRTE)
- Access Manager Web Security Environment
- Access Manager WebSEAL

► WebSphere Portal server node:

– IBM WebSphere Application Server Enterprise with the following features:

- WebSphere Application Server (Base) Version 5.0.2.3 (Version 5.0 with Fix Pack 2 and Cumulative Base Fix 3 and Fixes)
- Programming Module Enhancement (PME) Version 5.0.2.2 (Version 5.0 with Fix Pack 2 and Cumulative PME Fix 2)

– IBM WebSphere Portal Extend for Multiplatforms Version 5.0.2.1 (Version 5.0 with Fix Pack 2 and Cumulative Fix 1 and Fixes) with the following features:

- WebSphere Portal
- WebSphere Portal Content Publisher

– IBM DB2 UDB, Enterprise Server Edition Version 8.1.4.428 (Version 8.1 with Fix Pack 4a)

– IBM Java Runtime Environment (JRE) Version 1.3.1

– IBM Tivoli Access Manager for e-business Version 5.1.0.2
Access Manager Java Runtime Environment (PDJRTE)

- Lotus collaborative portlets
- Lotus Collaboration Center

► IBM Lotus Domino mail server:

  Lotus Domino Server Version 6.5.1

► IBM Lotus Team Workplace Server:

  - Lotus Domino Server Version 6.5.1
  - Lotus Team Workplace Version 6.5.1

► IBM Lotus Instant Messaging and Web Conferencing server:

  - Lotus Domino Server Version 6.5.1
  - Lotus Instant Messaging and Web Conferencing Version 6.5.1

3.1.3 Software installation source

There are approximately 80 CDs included in the IBM WebSphere Portal Extend for Multiplatforms V5.0.2. Due to the vast number IBM WebSphere Portal Extend for Multiplatforms V5.0.2 CDs and the not so obvious naming of the CDs, we have provided a list of the CDs we used in our environment for reference purposes in Table 3-1.

Table 3-1 IBM WebSphere Portal Extend for Multiplatforms V5.0.2 CDs

CD	WebSphere Portal component	Version
Setup	Portal Installer WebSphere Portal Toolkit Portal InfoCenter	Version 5.0 for Multiplatforms Version 5.0 Version 5.0.1
Fixpack	WebSphere Portal 5.0 Fix Pack	Fix Pack 2
1-1	WebSphere Application Server Enterprise for Windows	Version 5.0
1-6	WebSphere Application Server Fix Pack and Fixes for Windows and Linux®	Fix Pack 1 (V5.0.1)
1-17	WebSphere Application Server Fix Pack and Fixes	Fix Pack 2 (V5.0.2)
2	WebSphere Portal WebSphere Portal Content Publisher	Version 5.0 Version 5.0

In addition to the software included with the base software, the installation procedures provide instructions about how to download and install newer fix packs and fixes for the collaborative portal solution. 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 and Tivoli Access Manager V5.1.

3.1.4 Software installation paths and variables

Throughout this book, we use a shorthand notation to indicate a certain directory structure of the installation of the software. Table 3-2 lists the software installation paths and variables that we used in the implementation procedure.

Table 3-2   Software installation paths and variables

Software	Our install path	Variable
IBM DB2 UDB V8.1.2, Enterprise Server Edition	c:\sqllib	<db2_home>
IBM HTTP Server V1.3.26.2	c:\IBMHttpServer	<ihs_home>
IBM WebSphere Application Server V5.0.2.1	c:\WebSphere\AppServer	<was_home>
IBM WebSphere Portal V5.0.2.1	c:\WebSphere\PortalServer	<wp_home>
IBM Tivoli Access Manager WebSEAL V5.1.0.2	c:\ibm\Tivoli\PDWeb	<tws_home>
IBM Tivoli Access Manager V5.1.0.2	c:\ibm\Tivoli\tam	<tam_home>
IBM Tivoli Directory Server V5.2	c:\ibm\ldap	<tds_home>
IBM Java Runtime Environment (JRE) V1.3.1	c:\ibm\Java131	<jre_home>
IBM GSKit V7.0.1.16	c:\ibm\gsk7	<gsk7_home>

3.2 Collaborative portal interaction

This section discusses the interaction of the WebSphere Portal server with the Lotus Collaborative application and some security consideration involving its setup. Figure 3-2 on page 51 shows this connectivity.
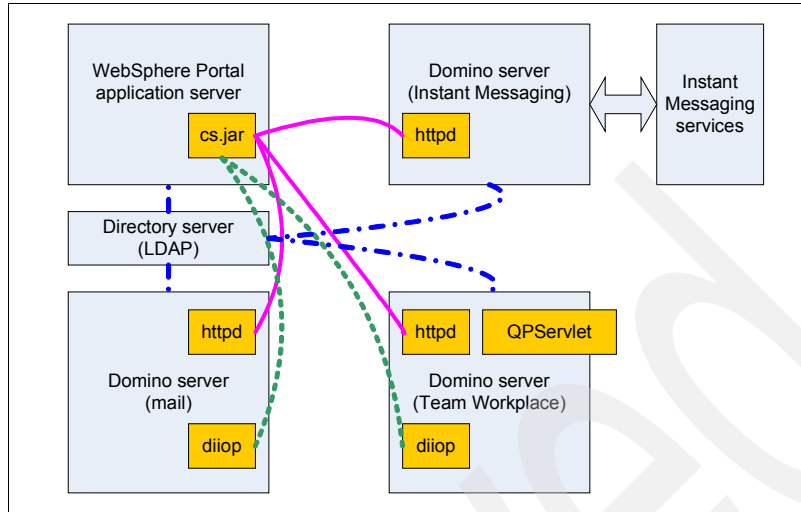
Figure 3-2 Component interaction and interconnection

In Figure 3-2, the connection is shown as lines between the components:

► The HTTP connection is shown with a straight line ——.

► The LDAP connection is shown with a dash-dot line —·—.

► The DIIOP connection is shown with a dotted line ····.

Most of the functions in the collaborative portlet for the Lotus collaboration application are performed using the Java classes contained in the cs.jar archive. This archive contains the necessary code to connect to Lotus Instant Messaging and Web Conferencing (formerly called Sametime), Lotus Team Workplace (formerly called QuickPlace), and Lotus Domino mail server.

The advanced functions of these application are performed using an internal frame or passing the content directly to the Lotus application HTTP daemon, so users need a connection directly to these services.

In this environment, security measures can be taken to provide security for the following aspects:

► Communication security: Because there are communication between the servers that should be secured, these communications are typically conducted using HTTP, LDAP, or DIIOP. These protocols can run over SSL.

► Integrated or separate identity management: The authentication of users is typically performed by a separate directory server. This can be conducted by a single directory server process or separately synchronized servers. All of these servers must have a consistent entry in the directories.

► Security token passing using an LTPA or TAI interaction.

► Reverse proxy consideration: There are several WebSEAL junctions that need to be defined. These are to the WebSphere Portal server, Lotus Instant Messaging and Web Conferencing, Lotus Team Workplace, and Lotus Domino Web Access (formerly called iNotes™).

3.2.1 Server picker overview

The server picker is automatically activated when you place one of the Domino portlets or QuickPlace Inline portlet in edit mode. The server picker issues an LDAP search to the host name and port specified in CSEnvironment.properties in the Domino Directory section. This **ldapsearch** command requests the name and http-hostname of every Domino server in the directory where the http-hostname field contains a value. If the http-hostname field does not contain a value, the server will not appear in the server picker list.

This task is completed on the Domino mail and application servers and enhances how users choose their mail or application database through the Domino portlets. Specifically, the DIIOP task must be correctly configured for the database picker feature to function. If you do not complete this step, users with access to edit the Domino portlets, Domino Web Access (formerly called iNotes), and QuickPlace Inline portlet will have to manually enter the database they need in the portlet, instead of being able to pick a database from the list in the picker as seen in the example. Also, it is important to point out that the database picker can only be accessed by users who have edit rights to the portlet itself. Depending on how you configure the security for the portlet, most end users will probably not have edit rights to the portlet, and therefore, never see the database picker at all.

> **Note:** Although the Lotus Domino Web Access, Notes Mail, My Inbox, My Calendar, and My ToDo portlets all use this functionality, they all first attempt to automatically detect the user's mail database. This feature to automatically detect the users mail database does not use DIIOP, but instead uses LDAP.

3.2.2 Overview of how the automatically detect my mail database feature works

The automatically detect a user's mail database feature is configured by default in all the Domino mail portlets, including Domino Web Access. When a user accesses a page with the desired mail portlet, the Collaborative Services will start two LDAP searches to the host name and port specified in CSEnvironment.properties in the Domino Directory section. The first LDAP search will send the user's login name pulled from the LTPA token generated in

portal when signing in, requesting the user's mail server and mail file fields from the user's Person document. This will return the user's mail file on the server (in our example, mail\iuser1.nsf) and the Domino canonical name of the user's mail server (in our example, kingston/itso). The Collaborative Services then issue another LDAP search to the host name and port specified in CSEnvironment.properties in the Domino Directory section, sending the Domino name just returned by the previous search, requesting the http-hostname of that Domino server (in our example, kingston.itsc.austin.ibm.com is returned). After it has these two values, the portlet can build the URL to which it must direct the user, the http://http-hostname/mail file (in our example, http://kingston.itsc.austin.ibm.com/mail/iuser1.nsf).

3.2.3 Database picker overview

The database picker attempts to populate a list of specific databases on a server after a user selects the desired server from the server picker, or manually enters the server host name and selects the check box. At this point, the collaborative servers attempt to connect to the Domino server specified by the user over IIOP. If the connection is made, a Java program attempts to find all databases based on a specific template. The template is based on the portlet in which you are using the picker.

► Domino Web Access: Displays databases using the Domino Web Access (iNotes) template.

► Mail portlets: Displays databases using the standard and extended Notes Mail template.

► My Lotus Notes View: Displays all Notes databases.

► My Lotus Notes Discussion: Displays databases using the Discussion template.

► My Lotus Notes Teamroom: Displays databases using the Teamroom template.

In order for the list of databases to display, the DIIOP task must be running, and the user attempting to use the picker must have access to run Java agents on the Domino server.

3.2.4 Lotus Team Workplace picker overview

The Lotus Team Workplace (formerly called QuickPlace) picker attempts to populate a list of places on a server after a user selects the desired server from the server picker, or manually enters the server host name and selects the check box. At this point, the collaborative servers attempt to connect to the Domino

server specified by the user over IIOP. If the connection is made, a Java program attempts to find all places on the server.

In order for the list of places to display, the DIIOP task must be running, and the user attempting to use the picker must have access to run Java agents on the Lotus Team Workplace server.

3.2.5 Portal awareness overview

When you sign in to WebSphere Portal, the Collaborative Services will look into the CSEnvironment.properties to see if Lotus Instant Messaging and Web Conferencing is enabled. If it is, it will authenticate you with the Instant Messaging server and build the stlinks applet into your browser. After you are authenticated with the Instant Messaging server, you should see the a new function called writeSTLinksApplet on every page source in WebSphere Portal, similar to the one in Example 3-1.

Example 3-1   Sample writeSTLinksApplet

```
    <script type="text/javascript" language="Javascript">
if (typeof writeSTLinksApplet == "function")
writeSTLinksApplet("uid=wpsadmin,cn=users,o=ibm,c=us", "<token written here>",
true);
</script>
```

The STLinksApplet handles all awareness in WebSphere Portal. If a portlet is enabled to show awareness, it will send names to show awareness to stlinks. Stlinks will, in turn, contact the Instant Messaging and Web Conferencing server directly to determine the user's status (active, away, do not disturb, or not online), and then will pass this information back to the portlet to show the user's status in the portlet.

If the STLinksApplet does not load after configuring single sign-on between WebSphere Portal and Lotus Instant Messaging and Web Conferencing, see the Technote *Troubleshooting Sametime Awareness in WebSphere Portal*, 1163790, available at:

    http://www.ibm.com/support/docview.wss?rs=0&uid=swg21163790

3.3 Implementation options

In this book, we discuss the implementation of the security options for the WebSphere Portal and Lotus Collaborative solution.

We start this discussion by outlining the necessary steps to implement the collaborative portal components using IBM Tivoli Directory Server and LTPA token passing. This is discussed in Chapter 4, "Implementing and configuring basic LTPA authentication with IBM Directory Server" on page 57.

Communication encryption and security for the components are achieved by activating SSL communication between the components. This is performed for HTTP, LDAP, and DIIOP links. In Chapter 5, "Setting up secure communication" on page 119, we discuss these steps.

Adding a front-end reverse proxy with IBM Tivoli Access Manager enables an additional layer of security and network partitioning, as described in Chapter 6, "Incorporating IBM Tivoli Access Manager for e-business" on page 167. This chapter also describes the implementation of LTPA and TAI with the reverse proxy.

In Chapter 7, "Integrating directory servers in an IBM WebSphere Portal environment" on page 221, we discuss an alternative approach to use to work with other directory servers, such as Domino LDAP and Microsoft Active Directory.

Implementing and configuring basic LTPA authentication with IBM Directory Server

This chapter describes how to install and configure the basic LTPA secure WebSphere Portal collaborative solution. This chapter serves as a base for the further security customization described in the following chapters. Our approach is to implement everything with only the necessary security, get this working, and then add additional security as needed. This chapter is organized into the following topics:

► Overview

► Implementing IBM WebSphere Portal

► Installing the Lotus Collaborative Components

► Installing Domino Extended Products portlets

► Placing portlets on a page for testing

► Known problems and fixes in this configuration

4.1 Overview

This section describes how to implement and configure the collaboration solution without a security solution. The overall structure for the scope of implementation in this chapter is shown in Figure 4-1.
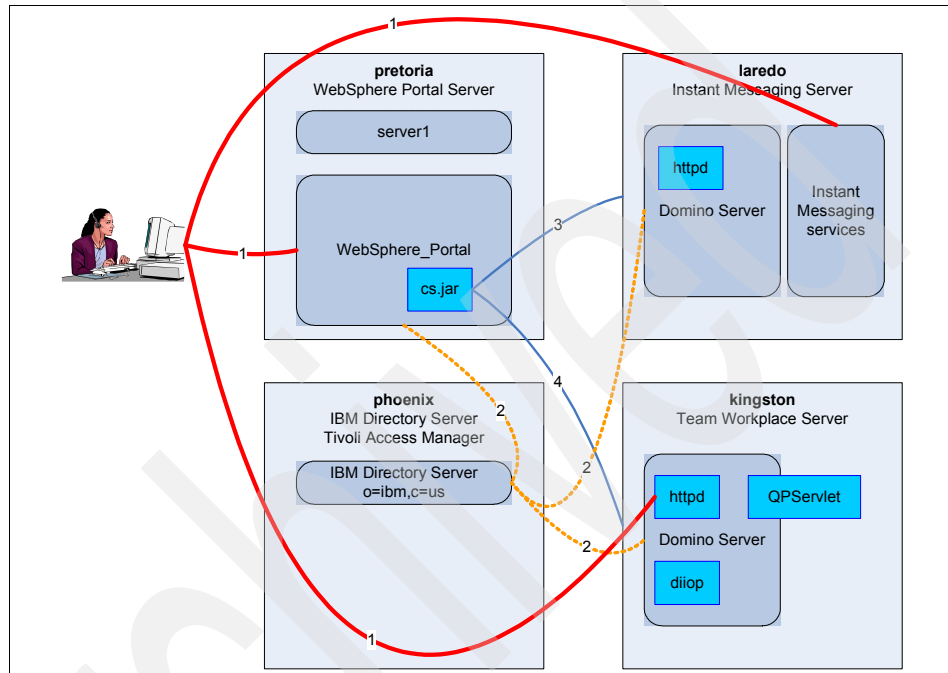
Figure 4-1   Implementation structure of the basic configuration

As discussed in Chapter 3, "Implementation planning and considerations" on page 45 and shown in Figure 4-1, our configuration is as follows:

► We did not use the reverse proxy node or the policy server node in this configuration.

► The directory server is using IBM Tivoli Directory Server in phoenix.

► The Domino mail server and Lotus Team Workplace server are merged in kingston.

► The Instant Messaging and Web Conferencing server is in laredo.

► The WebSphere Portal server node is implemented in pretoria.

The high-level tasks to install and configure this environment are as follows:

1. Install and configure WebSphere Portal with DB2 and IBM Directory Server, as explained in 4.2, "Implementing IBM WebSphere Portal" on page 59.

2. Install the Lotus software suites: Domino, Team Workplace, Instant Messaging and Web Conferencing, as discussed in 4.3, "Installing the Lotus Collaborative Components" on page 75.

3. Install the Domino Extended Products portlets, as discussed in 4.4, "Installing Domino Extended Products portlets" on page 87:

   a. Configure Domino to work with the Domino portlets.
   b. Configure the Lotus Team Workplaces to work with the Team Workplace portlets.
   c. Configure Lotus Instant Messaging and Web Conferencing to work with the Instant Messaging and Web Conferencing portlets.
   d. Configure People Finder to work with the IBM Directory Server LDAP directory.
   e. Configure Lotus Team Workplace to work with Lotus Instant Messaging and Web Conferencing.

4. Assign the portlet to the workspace of a user, as described in 4.5, "Placing portlets on a page for testing" on page 115.

4.2 Implementing IBM WebSphere Portal

This section describes the procedure we used to install and configure the WebSphere Portal server node for our example runtime environment on Microsoft Windows.

> **Note:** When installing and configuring WebSphere Portal, we referenced the following information:
>
> ► *IBM WebSphere Portal Extend for Multiplatforms V5.0.2 Information Center*, available at:
>
>   http://www.ibm.com/websphere/portal/library
>
> ► *IBM WebSphere Portal for Multiplatforms V5 Handbook*, SG24-6098
>
> ► *A Secure Portal Using WebSphere Portal V5 and Tivoli Access Manager V4.1*, SG24-6077
>
> ► *Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1*, SG24-6325

We describe the high-level tasks to install the WebSphere Portal server node in the following sections:

► Installing Base WebSphere Portal V5.0
► Upgrading WebSphere Portal to V5.0.2
► Upgrading to WebSphere Portal Cumulative Fix 1 (V5.0.2.1)
► Installing DB2 Universal Database
► Configuring WebSphere Portal for DB2
► Configuring WebSphere Portal for IBM HTTP Server
► Connecting WebSphere Portal to a directory server

4.2.1 Installing Base WebSphere Portal V5.0

This section describes how to install and configure WebSphere Portal Version 5.0. We installed the following components:

► WebSphere Application Server Enterprise V5.0.1, Base and Programming Module Enhancement (PME) including required fixes
► IBM HTTP Server V1.3.26
► WebSphere Portal V5.0
► WebSphere Portal Content Publishing V5.0

To install WebSphere Portal V5.0, we completed the following steps:

1. Insert the *IBM WebSphere Portal V5.0.2 Setup* CD. The installer will automatically start the installation process by offering a command prompt. If auto start is disabled, run install.bat from the root of the CD to start the installation.

2. In the Install Shield Language, select the desired language and click **OK**.

3. In the Welcome window, click **Next** to continue the installation.

4. In the License Agreement window, accept the terms in the license agreement and then click **Next**. The installer will check for the required operating system and prerequisites.

5. For the installation options, we choose **Full install**. Click **Next** to continue.

6. For the installation directory, we used C:\WebSphere\AppServer for WebSphere and C:\IBM\IBMHTTPServer for IBM HTTP Server. Click **Next**.

7. In prompts for the System Logon user ID and password that will be used to start the WebSphere Application Server and IBM HTTP Server programs, we start both as a service and use the administrator user ID and its password. Click **Next**.

8. Put in the node name and full host name for the WebSphere Portal server machine and click **Next**.

9. For the WebSphere Portal installation directory, we used C:\WebSphere\PortalServer. Click **Next**.

10. For the Portal administrative user and password. We used wpsadmin as the user ID. You can use any user name here. Click **Next**.

11. Confirm the different components that are going to be installed and click **Next**.

12. The installer program will then prepare the installation. After a while, you will be prompted to insert CD #1-1 *WebSphere Application Server Enterprise for Windows*. It will first start locating a Java Virtual Machine, and then begin the install of WebSphere Application Server (Base and PME).

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

    The wizard will then perform the following tasks, displaying a progress meter for each task:

    a. Prepare the WebSphere Application Server Fix Pack files.
    b. Install WebSphere Application Server Fix Pack 1.
    c. Install WebSphere Application Server Enterprise Fix Pack 1.
    d. Install WebSphere Application Server Fixes.

13. When these tasks are complete, the installer will start the WebSphere Application Server (server1 application server). After the server starts, you will be prompted to insert CD #2 *WebSphere Portal Server - WebSphere Portal Content Publisher*. The wizard will perform the following task:

    a. Install WebSphere Portal.
    b. Start WebSphere Portal server (WebSphere_Portal application server).
    c. Install the default portlets.
    d. Install the WebSphere Portal Content Publishing features.

    After the installation finishes, leave the check box selected for **Launch First Steps** and click **Finish**. The installation program will then complete and close. Note that as its final task, it will load the WebSphere Portal First Steps application.

If the installation completed properly, the WebSphere Portal First Steps application should be running. To verify that WebSphere Portal is working, complete the following steps:

1. Click **Launch WebSphere Portal** to test that the portal pages appear properly. In our case, this launched a Web browser window with the following URL:

```
http://pretoria.itsc.austin.ibm.com:9081/wps/portal
```

2. Log in to the portal by clicking the **Log in** link located in the upper-right corner of the page. This will take you to a new page prompting for login information. Use the administrative user ID and its password that you entered in step 10 on page 61.

3. You should be presented with the personalized welcome portal page for the logged in user.

4. Click **Log out**.

4.2.2 Upgrading WebSphere Portal to V5.0.2

IBM WebSphere Portal Extend for Multiplatforms V5.0.2 requires IBM WebSphere Application Server Enterprise V5.0.2. There are several components in IBM WebSphere Application Server Enterprise V5.0.2: the IBM WebSphere Application Server (Base), WebSphere Application Server Network Deployment, WebSphere MQ, and Programming Module Enhancements (PME).

The WebSphere Application Server Enterprise V5 Fix Pack 2 requires write access to the file system during installation. For this reason, we need to copy the fix pack to the local file system of the target node. In addition, Fix Pack 2 includes a newer version of the WebSphere Update Installer (install wizard for fix packs and fixes). We use the WebSphere Update Installer included with Fix Pack 2 to install the WebSphere fixes.

On top of the WebSphere Application Server Enterprise V5 Fix Pack 2, there are some additional fixes that are required by WebSphere Portal V5.0.2. These are:

► PQ76567_5.0.2.jar
► PQ78166eFix_fixes_install_db_resource.jar
► PQ81248_fix.jar
► WAS_CM_08-12-2003_5.0.2-5.0.1_cumulative_Fix.jar
► PQ75469.jar
► PQ77008.jar
► PQ77142.jar
► PQ78370_Fix.jar
► PQ78382_fix.jar
► PQ78882_Fix.jar
► PQ79083_5.0.2_Fix.jar
► PQ79193_fix.jar
► PQ81020_fix.jar
► WAS_Adapter_10-30-2003_5.0.2_cumulative_Fix.jar
► WAS_Sessions_08-12-2003_5.0.2_cumulative_Fix.jar
► WAS_Security_07-07-2003_5.0.2-5.0.1-5.0.0_JSSE_cumulative_Fix.jar
► WAS_Plugin_09-03-2003_5.0.X_cumulative_Fix_Win.jar

These fixes are supplied in the *WebSphere Application Server Fixpack and eFixes for Windows and Linux* CD.

Complete the following steps to install the WebSphere Application Server Enterprise V5 Fix Pack 2 (V5.0.2):

1. Stop the servers and back up the configuration.

   Prior to starting the cumulative fix installation, ensure that the server Windows services are stopped, and back up the WebSphere Application Server configuration as follows:

   a. Ensure that the following servers are stopped before you install Fix Pack 2:

      • All application servers including server1 and WebSphere_Portal; check using the `serverStatus -all` command.
      • IBM HTTP Server.
      • IBM HTTP Administration Server.

   b. Back up the WebSphere Application Server configuration by entering the command `backupConfig` from c:\WebSphere\AppServer\bin\.

2. Install WebSphere Application Server V5 Fix Pack 2.

   To install WebSphere Application Server V5 Fix Pack 2 (V5.0.2), complete the following steps on the WebSphere Portal server node:

   a. Copy the WebSphere Application Server Base Fix Pack 2 files and subdirectories found in the <CD_Root>\wasfp2\win directory of the *WebSphere Application Server Fixpack and eFixes for Windows and Linux* CD to a temporary directory (for example, c:\temp\was5.fp2).

   b. Set up the environment by running the `setupCmdLine.bat` command.

   c. To start the WebSphere Update Installer supplied with WebSphere Application Server Base Fix Pack 2, we run the `updateWizard.bat` command from C:\temp\was5.fp2. Follow the installation wizard, and select the following options:

      • Select **Install Fix Packs**.
      • The source path is `C:\temp\was5.fp2\fixpacks`.
      • Select the **was50_fp2_win** Fix Pack.

   d. When the WebSphere Application Server V5 Base Fix Pack 2 installation is complete, click **Finish**.

3. Install WebSphere Application Server PME V5 Fix Pack 2.

To install WebSphere Application Server Programming Module Extension (PME) V5 PME Fix Pack 2 (V5.0.2), complete the following steps:

a. Copy the WebSphere Application Server PME Fix Pack 2 files and subdirectories found in the <CD_Root>\pmefp2\win directory of the *WebSphere Application Server Fixpack and eFixes for Windows and Linux* CD to a temporary directory (for example, c:\temp\was5pme.fp2).

b. Set up the environment by running the `setupCmdLine.bat` command.

c. To start the WebSphere Update Installer supplied with WebSphere Application Server PME Fix Pack 2, we run the `updateWizard.bat` command from C:\temp\was5pme.fp2. Follow the installation wizard, and select the following options:

- Select **Install Fix Packs**.
- The source path is `C:\temp\was5pme.fp2\fixpacks`.
- Select the **was50_pme_fp2_win** Fix Pack.

d. When the WebSphere Application Server V5 PME Fix Pack 2 installation is complete, click **Finish**.

4. Install WebSphere Application Server V5.0.2 Fixes.

To install WebSphere Application Server V5.0.2 eFixes, complete the following steps on the WebSphere Portal server node:

a. Copy the WebSphere Application Server eFixes files and subdirectories found in the <CD_Root>\fixes\win directory of the *WebSphere Application Server Fixpack and eFixes for Windows and Linux* CD to a temporary directory (for example, c:\temp\was502.efixes).

b. Set up the environment by running the `setupCmdLine.bat` command.

c. To start the WebSphere Update Installer, we run the `updateWizard.bat` command from C:\temp\was502.efixes. Follow the installation wizard, and select the following options:

- Select **Install Fixes**.
- The source path is `C:\temp\was502.efixes\efixes`.
- Select all the fixes from the list.

d. When the WebSphere Application Server V5.0.2 Fixes installation is complete, click **Finish**.

Prior to continuing to the WebSphere Portal V5 Fix Pack 2 installation, we recommend that you verify that the WebSphere Application Server V5.0.2 is working properly.

To install the WebSphere Portal V5 Fix Pack 2 (V5.0.2), complete the following steps:

1. Ensure that all the application servers are stopped before you start the installation. To check the server status, enter the `serverStatus -all` command. If you receive a message that none of the servers can be reached, they are all stopped. If one or more servers display as running, you can use the `stopServer <servername>` command to stop the server.

2. Copy the WebSphere Portal V5 Fix Pack 2 files and subdirectories found in the CD to a temporary directory (for example, c:\temp\wps502).

3. Set up the environment by running the `setupCmdLine.bat` command.

4. From the WebSphere Portal update directory, c:\temp\wps502, start the WebSphere Portal update installer. Enter the following command:

   ```
   updatePortal -fixpack -installDir c:\WebSphere\PortalServer -fixpackDir
   c:\temp\wps502 -install -fixpackID WP_PTF_502
   ```

   Note: If you configured WebSphere Application Server V5 security prior to installing the WebSphere Portal V5 Fix Pack 2, you will need to ensure that the WasUserid and WasPassword fields are populated correctly in the <wp_home>\config\wpconfig.properties file before running the WebSphere Portal V5 Fix Pack 2 update installer.

   When the fix pack installation is complete, you should see the following message:

   ```
   Fix pack installation completed successfully.
   ```

5. Update the WebSphere Portal configuration. After the WebSphere Portal V5 Fix Pack installation is complete, you will need to update the WebSphere Portal configuration. For more information, see:

   ```
   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/readme/install_win_unix.html
   ```

   a. Restart the IBM HTTP Server.

   b. Open a command window and enter the following command to update the WebSphere Portal configuration:

      ```
      c:\WebSphere\PortalServer\config\WPSConfig.bat WP-PTF-502
      -DPortalAdminPwd=<password>
      ```

      Where <password> is the WebSphere Portal administrator password.

There are some additional components of WebSphere Portal that can be upgraded to Fix Pack 2 level, but these components are not relevant to our example here. These components are:

► IBM WebSphere Portal Content Publisher V5
► Document Manager search index

Now that you have installed and configured WebSphere Portal V5 Fix Pack 2 (V5.0.2), we recommend that you verify that WebSphere Portal Server is working properly.

4.2.3 Upgrading to WebSphere Portal Cumulative Fix 1 (V5.0.2.1)

IBM WebSphere Portal V5.0.2 Cumulative Fix 1 (V5.0.2.1) requires the following levels:

► WebSphere Application Server V5.0.2 Base Cumulative Fix 3 (V5.0.2.3)
► WebSphere Application Server V5.0.2 PME Cumulative Fix 2 (V5.0.2.2)

WebSphere Application Server V5.0.2 Base Cumulative Fix 2 must be installed before installing the PME Cumulative Fix 2. Then, you can install WebSphere Application Server V5.0.2 Cumulative Fix 3 (V5.0.2.3). In addition, WebSphere Portal V5.0.2.1 requires several WebSphere Application Server V5.0.2.3 eFixes.

> **Important:** WebSphere Portal does not work after installing the WebSphere Application Server fix packs and fixes. It will work again after installing the WebSphere Portal Cumulative Fix 1 (V5.0.2.1).

Based on the previous requirement, the installation is divided into the following steps:

1. Stop the servers and back up the configuration.

   Prior to starting the cumulative fix installation, ensure that the server Windows services are stopped, and back up the WebSphere Application Server configuration as follows:

   a. Ensure that the following servers are stopped before you install Fix Pack 2:
      • All application servers, including server1 and WebSphere_Portal. Check using the `serverStatus -all` command.
      • IBM HTTP Server.
      • IBM HTTP Administration Server.

   b. Back up the WebSphere Application Server configuration by entering the `backupConfig` command from c:\WebSphere\AppServer\bin\.

2. Install WebSphere Application Server V5.0.2 Cumulative Fix 2 (V5.0.2.2):

   a. Download the WebSphere Application Server V5.0.2 Cumulative Fix 2 (V5.0.2.2) for Windows, was502_cf2_win.zip, from the following URL:

      http://www.ibm.com/support/docview.wss?rs=203&context=SW000&uid=swg24005954

   b. Unpack the ZIP file into a temporary directory, for example, C:\temp\was502.cf2.

c. Set up the environment by running the **setupCmdLine.bat** command.

d. Run the **updateWizard.bat** command from C:\temp\was502.cf2. Follow the installation wizard, and select the following options:

- Select **Install Fix Packs**.
- The source path is `C:\temp\was502.cf2\fixpacks`.
- Select the **was502_cf2_win** Fix Pack.

e. When the installation is complete, click **Finish**.

3. Install WebSphere Application Server PME V5.0.2 Cumulative Fix 2 (V5.0.2.2):

a. Download the WebSphere Application Server PME V5.0.2 Cumulative Fix 2 (V5.0.2.2) for Windows, was502_pme_cf2_win.zip, from the following URL:

`http://www.ibm.com/support/docview.wss?rs=823&context=SS4QY3&uid=swg2400 5954`

b. Unpack the ZIP file into a temporary directory, for example, C:\temp\was502pme.cf2.

c. Set up the environment by running the **setupCmdLine.bat** command.

d. Run the **updateWizard.bat** command from C:\temp\was502pme.cf2. Follow the installation wizard, and select the following options:

- Select **Install Fix Packs**.
- The source path is `C:\temp\was502pme.cf2\fixpacks`.
- Select the **was502_pme_cf2_win** Fix Pack.

e. When the installation is complete, click **Finish**.

4. Install WebSphere Application Server V5.0.2 Cumulative Fix 3 (V5.0.2.3):

a. Download the WebSphere Application Server V5.0.2 Cumulative Fix 3 (V5.0.2.2) for Windows, was502_cf3_win.zip.

b. Unpack the ZIP file into a temporary directory, for example, C:\temp\was502.cf3.

c. Set up the environment by running the **setupCmdLine.bat** command.

d. Run the **updateWizard.bat** command from C:\temp\was502.cf3. Follow the installation wizard, and select the following options:

- Select **Install Fix Packs**.
- The source path is `C:\temp\was502.cf3\fixpacks`.
- Select the **was502_cf3_win** Fix Pack.

e. When the installation is complete, click **Finish**.

5. Install WebSphere Application Server V5.0.2.3 eFixes:

   a. Download the WebSphere Application Server V5.0.2 Cumulative Fix 2 (V5.0.2.2) for Windows, WAS5023CumulativeWindows.zip, from the following URL:

      http://www.ibm.com/support/docview.wss?uid=swg24006309

   b. Unpack the ZIP file into a temporary directory, for example, C:\temp\was5023.fixes.

   c. Set up the environment by running the **setupCmdLine.bat** command.

   d. Run the **updateWizard.bat** command from C:\temp\was5023.fixes. Follow the installation wizard, and select the following options:

      • Select **Install Fixes**.
      • The source path is `C:\temp\was5023.fixes\fixes`.
      • Select *only* the fixes shown in Table 4-1.

   e. When the installation is complete, click **Finish**.

Table 4-1   WebSphere Application Server V5.0.2.3 fixes

Fix description
WAS_Dynacache_01-30-2004_5.0.2_cumulative
PQ78370
PQ81248
PQ81416
WAS_Security_12-12-2003_5.0.2.3-5.0.2.2-5.0.2.1-5.0.2-5.0.1-5.0.0_JSSE_cumulative_Fix
WAS_CM_08-12-2003_5.0.2-5.0.1_cumulative_Fix
WAS_Adapter_10-30-2003_5.0.2_cumulative_Fix
WebSphere Plug-in Cumulative Fix for 5.0.0, 5.0.1, and 5.0.2

Prior to continuing to the WebSphere Portal V5 Cumulative Fix 1 installation, we recommend that you verify that the WebSphere Application Server V5.0.2.3 is working properly.

Note: WebSphere Portal will not work with the level of WebSphere Application Server at this stage. WebSphere Portal will work after applying the WebSphere Portal Cumulative Fix 1 (V5.0.2.1).

To install the WebSphere Portal V5.0.2 Cumulative Fix 1 (V5.0.2.1), complete the following steps:

1. Ensure that all the application servers are stopped before you start the installation. To check the server status, enter the `serverStatus -all` command. If you receive a message that none of the servers can be reached, they are all stopped. If one or more servers display as running, you can use the `stopServer <servername>` command in order to stop the server.

   Note: If you configured IBM HTTP Server or DB2, you might need to perform additional actions before installing Cumulative Fix 1.

2. Download the following to the <wp_root>\update directory (for example, c:\websphere\portalserver\update) on the WebSphere Portal server node:

   ```
   http://www.ibm.com/support/docview.wss?rs=688&context=SSHRKX&q1=cumulative+
   fix&uid=swg24006865&loc=en_US&cs=utf-8&lang=en+en
   ```

   – WebSphere Portal Update Installer (PortalUpdateInstaller.zip). This is required to install the WebSphere Portal V5.0.2 Cumulative Fix 1.

   – WebSphere Portal V5.0.2 Cumulative Fix 1 (V5.0.2.1) for Windows (WP_PTF_5021.jar).

   Note: From the URL listed, you will have to log in as a registered user (or register first). After you navigate to the download page, you will see a list of many fixes. We downloaded the following:

   ► WebSphere Portal Update Installer (PortalUpdateInstaller.zip)
   ► Portal 5.0.2 Cumulative Fix 1 (WP_PTF_5021.jar)

3. Unzip the contents of PortalUpdateInstaller.zip to the <wp_root>\update directory (overwriting any current files if necessary).

4. Change the time out for the SOAP client (if you have not done so previously):

   a. Open the soap.client.props file in <was_home>\properties.

   b. Modify the request time out as follows (the default is 180 seconds):

   ```
   com.ibm.SOAP.requestTimeout=6000
   ```

   c. Save and close the file.

5. Open a command window and enter the following command to set up the Java environment for the WebSphere Portal update installer:

   ```
   C:\WebSphere\AppServer\bin\setupCmdLine.bat
   ```

6. Navigate to the WebSphere Portal update directory:

   ```
   cd \WebSphere\PortalServer\update
   ```

7. To start the WebSphere Portal update installer, enter the following command:

```
updatePortal -fixpack -installDir c:\WebSphere\PortalServer -fixpackDir
c:\WebSphere\PortalServer\update -install -fixpackID WP_PTF_5021
```

When the fix pack installation is complete, you should see the following message:

```
Fix pack installation completed successfully.
```

If you receive an error, you can review the log information in the c:\WebSphere\PortalServer\log directory.

8. Restart the IBM HTTP Server.

9. Open a command window and enter the following command to update the WebSphere Portal configuration:

```
c:\WebSphere\PortalServer\config\WPSConfig.bat WP-PTF-5021
-DPortalAdminPwd=<password>
```

Where *<password>* is the WebSphere Portal administrator password.

> **Note:** For details, refer to the *WebSphere Portal V5 Cumulative Fix readme*
> (install_win_unix.html), available at:
>
> http://publib.boulder.ibm.com/pvc/wp/5021/ent/en/readme/install.html

10. Restart WebSphere Application Server:

```
cd \websphere\appserver\bin
stopServer server1
startServer server1
```

11. Restart WebSphere Portal server:

```
stopServer WebSphere_Portal
startServer WebSphere_Portal
```

Now that you have installed and configured WebSphere Portal V5 Cumulative Fix 1 (V5.0.2.1), we recommend that you verify that the WebSphere Portal Server is working properly.

4.2.4 Installing DB2 Universal Database

In this section, we describe how to install the IBM DB2 Universal Database V8.1, Enterprise Server Edition and supporting Fix Pack 4a. In our example, DB2 was installed on the same machine as WebSphere Portal.

Complete the following steps:

1. Install DB2 UDB V8.1. From the *DB2 UDB V8.1 Enterprise Server Edition* CD, run the `setup.exe` command from the CD and follow the installation wizard. Select the following options:
   - Select the **Typical** installation.
   - Select the Installation Folder; we entered `c:\ibm\SQLLIB`.
   - We use the DB2 administrator user db2admin.

2. Install DB2 UDB V8.1 Fix Pack 4a. This is the level that we chose to use. You can consult the product documentation for other DB2 levels. The IBM DB2 UDB V8.1 Fix Pack 4a, FP4a_WR21338_ESE.exe, can be downloaded from:

   http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8fphist.d2w/report#WIN-32

   Run FP4a_WR21338_ESE.exe to install the IBM DB2 UDB V8.1 Fix Pack 4a. We accepted the default installation options. We recommend that you restart your system after installing the fix pack to ensure that all fixes are applied and active in memory. After the system has restarted, open a DB2 command window (or Windows command window) and enter the `db2level` command, which should return level 8.1.4.428.

3. Verify that DB2 UDB is running.

4.2.5 Configuring WebSphere Portal for DB2

In this section, we describe how to configure and verify WebSphere Portal V5.0.2.1 to use DB2 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. Open a command window and navigate to the <wp_home>\config directory.

2. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file by running the `wpsconfig backup-main-cfg-file` command. You should get a backup of the wpconfig.properties file with a time stamp in the <wp_home>\config directory.

> **Note:** The backup configuration procedure should be run prior to all configuration tasks, such as configuring security, externalizing the Web server, or transferring the database (as is the case for our example).
>
> The wpconfig.properties file is a configuration input file used by wpsconfig to load configuration settings for WebSphere Portal (stored in XML files).

3. Export the WebSphere Portal configuration data found in the Cloudscape database by entering the following command:

```
wpsconfig database-transfer-export
```

When this completes, you should get the message BUILD SUCCESSFUL.

4. Modify the wpconfig.properties file to configure WebSphere Portal to use DB2 UDB. Refer to Table 4-2 for the configuration settings used in our example. For a detailed description of each of the keywords, refer to the *WebSphere Portal Information Center*, available at:

http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_db2.html

Table 4-2   WebSphere Portal configuration settings in the wpconfig.properties file for DB2

Section of wpconfig.properties file	Keyword	Our 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	db2admin
	DbPassword	<your_dbuser_password>
	DbLibrary	c:/ibm/sqllib/java/db2java.zip
	WpsDsName	wps50DS
	WpsXDbName	wps5TCP
	WpsDbNode	wpsNode

Section of wpconfig.properties file	Keyword	Our value
WebSphere Portal content publishing Database properties	WpcpDbNode	wcmNode
	WpcpXDbName	wpcp5TCP
	FeedbackXDbName	fdbk5TCP
	WpcpDbName	wpcp50
	WpcpDbUser	db2admin
	WpcpDbPassword	<your_dbuser_password>
	WpcpDbUrl	jdbc:db2:wpcp50
	WpcpDbEjbPassword	<ejb_password> (user defined)
	FeedbackDbName	fdbk50
	FeedbackDbUser	db2admin
	FeedbackDbPassword	<your_dbuser_password>
	FeedbackDbUrl	jdbc:db2:fdbk50
Member Manager properties	WmmDsName	wmmDS
	WmmDbName	wps50
	WmmDbUser	db2admin
	WmmDbPassword	<your_dbuser_password>
	WmmDbUrl	jdbc:db2:wps50

5. Save the updated wpconfig.properties file.

6. Create the databases. If you are using a local DB2 database, you can automatically create your databases using the command:

```
WPSconfig.bat create-local-database-db2
```

7. Validate the configuration properties. From the <wp_home>/config directory, enter the following commands to validate the configuration properties:

```
WPSconfig.bat validate-database-connection-wps
WPSconfig.bat validate-database-connection-wmm
WPSconfig.bat validate-database-connection-wpcp
WPSconfig.bat validate-database-driver
```

8. Import the database settings. If the validation runs correctly, enter the **WPSconfig.bat database-transfer-import** command to run the configuration task to import the database settings.

9. After importing the database tables, perform a reorg check to improve performance. Use a DB2 command window and perform the following commands for WebSphere Portal databases (for example, wps50, wpcp50, and fdbk50):

```
db2 connect to <database_name>
db2 reorgchk update statistics on table all
db2 terminate
db2rbind <database_name> -l db2rbind.out -u db2admin -p <password>
```

10. Start the WebSphere Portal server as follows:

```
cd \ibm\WebSphere\AppServer\bin
startServer WebSphere_Portal
```

11. Verify the WebSphere Portal for DB2 configuration.

4.2.6 Configuring WebSphere Portal for IBM HTTP Server

We now reconfigure WebSphere Portal 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, complete the following steps:

1. Verify that the IBM HTTP Server is started.

2. Navigate to the <wp_home>\config directory.

3. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file by entering the following command:

```
wpsconfig backup-main-cfg-file
```

4. Change the wpconfig.properties values as shown in Table 4-3 on page 75. For a detailed description of each of the keywords, refer to the *WebSphere Portal Information Center*, available at:

http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/inst_ihs.html

Table 4-3   Our example wpconfig.properties values for the external Web server

Section in wpconfig.properties file	Keyword	Our example value
WebSphere Application Server	WpsHostName	pretoria.itsc.austin.ibm.com
	WpsHostPort	80

5. Save the updated wpconfig.properties file.

6. Enter the `wpsconfig httpserver-config` command to configure WebSphere Portal for the external Web server.

7. Restart the IBM HTTP Server and restart the WebSphere Portal server.

8. 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, which in our example is:

   `http://pretoria.itsc.austin.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, which for our example is:

   `http://pretoria.itsc.austin.ibm.com:9081/wps/portal`

   Now that we have configured the external Web server, we do not need to specify the port (default port 80) in the URL we use for our example:

   `http://pretoria.itsc.austin.ibm.com/wps/portal`

4.2.7 Connecting WebSphere Portal to a directory server

In Chapter 7, "Integrating directory servers in an IBM WebSphere Portal environment" on page 221, we discuss the directory server configuration. Specific instructions for installing and configuring IBM Tivoli Directory Server Version 5.2 is provided in 7.1, "IBM Tivoli Directory Server V5.2 environment" on page 222.

4.3 Installing the Lotus Collaborative Components

This section explains how to install and configure the Domino servers. There are three types of Domino install processes that can be performed:

► Domino Base server: This is the basic installation of Domino. This is used as the base installation of the Domino server before installing Lotus Team Workplaces and Lotus Instant Messaging and Web Conferencing servers.

- Domino LDAP server: You only need one Domino server in your environment running the LDAP task, but for the Domino portlets and the QuickPlace Inline portlet, you will need this server installed and configured.

- Domino mail or application server: You can have any number of Domino mail or application servers in your environment.

For our example, we install three Domino servers:

- Domino LDAP server: warsaw

- Domino mail and application server and the Team Workplace server: kingston

- Domino server for Lotus Instant Messaging and Web Conferencing server: laredo

We discuss the installation processes in the following sections:

- Installing Lotus Domino V6.5.2

- Installing Lotus Team Workplace V6.5.1

- Installing Lotus Instant Messaging and Web Conferencing

- Activating server processes for Domino

4.3.1 Installing Lotus Domino V6.5.2

In this section, we take you through the steps to install a basic Domino server. We install the most basic installation of Domino; only the absolute necessary tasks are left running. Later, when we configure the servers to work with WebSphere Portal, we will turn on the tasks that could have been enabled during the install. We did this for two reasons. First, if you already have your Domino servers installed, you can skip this step and not miss enabling any Domino tasks necessary to work with WebSphere Portal. Second, we believe that by enabling the tasks on the Domino server only when necessary, this will help explain how certain features work and let you choose what you want and do not want to run on Domino.

For more information about installing Lotus Domino and all the Domino tasks, see the *Lotus Domino Administration Help*, available at:

    http://www.lotus.com/ldd/notesua.nsf/find/domino

The Domino server code is installed at all Domino servers (warsaw, kingston, and laredo). Run `setup.exe` from the Domino 6.5.2 CD and follow the installation wizard. We used the following options:

- We did not select Partitioned Server Install.

- We used the defaults for the installation folders: C:\Lotus\Domino and C:\Lotus\Domino\Data.

► We choose **Domino Enterprise Server** as the installation type.

Next, we need to configure the server. There are two types of Domino server configurations: for the first Domino server and for additional Domino servers. In our environment, kingston is the first Domino server, and warsaw and laredo are configured as additional Domino servers.

For the first Domino server, kingston, we performed the following steps:

1. Start the server by selecting **Programs** → **Lotus Applications** → **Lotus Domino Server**.

2. In the Welcome to Domino Server Setup window, click **Next**.

3. In the First or additional server window, select **Set up the first server** or a **stand-alone server**, and then click **Next**.

4. In the Provide a server name and title window, enter your Server name (in our case, kingston) and a title if you want. Click **Next**.

5. In the Choose your organization name window, enter an Organization name (in our case, itso) and type a password. Click **Next**.

6. In the Choose the Domino domain name window, enter a Domino domain name (in our case, itso) and click **Next**.

7. In the Specify an Administrator name and password window, enter a Domino administrator name and password (Domino Admin in our example) and click **Next**.

8. Clear all of the options under **Setup Internet services for**. We will configure these as needed. Click **Next**.

9. In the Domino network settings window, click the **Customize** button.

   a. Ensure that the **TCP/IP** option is selected.

   b. Under Type the fully qualified internet host name for this Domino server, we entered kingston.itsc.austin.ibm.com.

   c. Click **OK**.

10. Back in the Domino network setting window, click **Next**.

11. In the Secure your Domino Server window, leave the defaults and click **Next**.

12. Click **Setup** to configure your Domino server.

13. Click **OK** in the Successful configuration window.

14. Start the Domino server by selecting **Programs** → **Lotus Applications** → **Lotus Domino Server**.

15. Install and configure the Notes Administration Client.

   a. Start the Notes Administration Client configuration by selecting **Programs → Lotus Applications → Lotus Notes**.

   b. Click **Next** in the Welcome window.

   c. In the User Information window:

      • For Your Name, enter the Domino administrator name you entered in step 7 on page 77.

      • For Domino server, enter the Domino server you created in <xref>. This name is a combination of the server name and the organization you create in those steps (kingston/itso in our example)

      • Ensure that the **I want to connect to a Domino server** option is selected.

      Click **Next**.

   d. Enter the password you chose in step 7 on page 77 and click **OK**.

   e. Clear the **Setup instant messaging** option in the Instant Messaging Setup window. Click **Next**.

   f. Under Additional Services, leave all the options cleared. Click **Finish**.

   Your Notes client should start automatically.

For additional Domino servers, complete the following configuration procedure:

1. Start the server by selecting **Start → Programs → Lotus Applications → Lotus Domino Server**.

2. Click **Next** in the Welcome to Domino Server Setup window.

3. In the First or additional server window, select **Set up an additional server**, and then click **Next**.

4. Select **The server ID file is stored in the Domino Directory** option, and enter the password. Click **Next**.

5. In the Provide the registered name of this additional Domino server window, enter this additional server name (in our case, laredo/itso) and click **Next**.

6. Clear all of the options under **Setup Internet services for**. We will configure these as needed. Click **Next**.

7. Click the **Customize** button in the Domino network settings window.

   a. Ensure that the **TCP/IP** option is not selected.

   b. Under Type the fully qualified internet host name for this Domino server, make sure that it is the fully qualified Internet host name for this server (in our case, laredo.itsc.austin.ibm.com).

c. Click **OK**.

8. When you return to the Domino network settings window, click **Next**.

9. In the Provide the system databases for this Domino server window:

    a. For Other Domino server name, type your first Domino server's name (kingston/itso in our example)

    b. For Optional network address, type the fully qualified Internet host name for that server (in our case, kingston.itsc.austin.ibm.com)

    c. Click **Next**.

10. Select **Set up as a primary Domino Directory (Recommended)** and click **Next**.

11. In the Secure your Domino Server window, leave the defaults and select **Next**.

12. Click **Setup** to configure your Domino server.

13. Click **OK** in the Successful configuration window.

14. Start the Domino Server by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Domino Server**.

After the server starts successfully, you might want to clear the password from the IB file, as explained in "Clearing the password on an additional server's ID file" on page 85.

4.3.2 Installing Lotus Team Workplace V6.5.1

In this section, we take you through the steps to install a Lotus Team Workplace (formerly called QuickPlace) server. The Lotus Team Workplace server installs on top of a Domino server, so prior to installing Lotus Team Workplace, make sure that you complete the steps in 4.3.1, "Installing Lotus Domino V6.5.2" on page 76 for the Team Workplace server.

To install Lotus Team Workplace V6.5.1, complete the following steps:

1. Stop the Domino server on which Lotus Team Workplace will be installed.

2. Run `setup.exe` from the Lotus Team Workplace CD.

3. Accept the license agreement.

4. Verify that the installation directories match the Domino server directories and allow the installation to continue.

5. When prompted, enter the name and password for a Team Workplace administrator. This user name and password is local to the Team Workplace server, and should *not* be the same user as anyone listed in the Domino Directory on which Team Workplace is installed or the LDAP directory with which Team Workplace will work. (We used qpadmin in our example.)

6. When prompted, click **Finish** to complete the installation.

7. Open the Notes.ini file from the Team Workplace server in a text editor. The Notes.ini file is located in the Domino program directory.

8. The Team Workplace server loads when the HTTP task loads in Domino. After installing Team Workplace, ensure that the HTTP task is set to automatically load in Domino. Find the `ServerTasks=` line and add `,http` to the end if it does not already exist.

9. Restart the Team Workplace server by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Domino Server**.

When the HTTP task loads, you should see the text shown in Example 4-1 in the Domino Console.

Example 4-1   Console of Team Workplace starting successfully

```
08/26/2004 02:31:25 PM  HTTP Server: DSAPI Domino Off-Line Services HTTP
extension Loaded successfully
08/26/2004 02:31:25 PM  HTTP Server: DSAPI QuickPlace DSAPI Filter Loaded
successfully
08/26/2004 02:31:27 PM  QuickPlace Server started. 350172.00
08/26/2004 02:31:29 PM  HTTP Server: Started
```

To enable Lotus Team Workplace to work with IBM Tivoli Directory Server, see 7.1.4, "Configuring Team Workplace with IBM Tivoli Directory Server" on page 236.

4.3.3 Installing Lotus Instant Messaging and Web Conferencing

In this section, we take you through the steps to install the Lotus Instant Messaging and Web Conferencing (formerly called Sametime) server. The Lotus Instant Messaging and Web Conferencing server installs on top of a Domino server, so prior to installing Lotus Team Workplace, ensure that you complete the steps in 4.3.1, "Installing Lotus Domino V6.5.2" on page 76 for the Team Workplace server.

Important: Do not install the Instant Messaging and Web Conferencing server and Team Workplace server on the same Domino server. These should be two separate Domino servers.

To install Lotus Instant Messaging and Web Conferencing, complete the following steps:

1. Stop the Domino server on which Lotus Instant Messaging and Web Conferencing will be installed.

2. Run `setup.exe` on the Instant Messaging and Web Conferencing CD.

3. Accept the license agreement.

4. Verify that the installation directories match the Domino server directories and allow the installation to continue. When prompted, click **Finish**.

5. When prompted, browse to and select the server.id file C:\Lotus\Domino\Data\server.id.

6. Select **LDAP** from the drop-down list and enter the host name and port in the required fields (phoenix.itsc.austin.ibm.com and 389 in our example).

7. In the HTTP tunnelling window, select the option to allow HTTP tunnelling.

> **Note:** If you are planning to integrate with Tivoli Access Manager, you must enable HTTP tunneling by selecting this option. If, however, you are not going to use Tivoli Access Manager or any other type of reverse proxy, you can leave this option cleared, and users will need to have access to the Instant Messaging and Web Conferencing server through these default ports:
>
> ► 1533: For Instant Messaging
> ► 8082: For chat
> ► 8081: For Web Conferences
> ► 554: For real-time streaming

8. Do not configure Lotus Instant Messaging and Web Conferencing to be managed by an Enterprise Meeting Server.

9. Click **Finish** to complete the installation of Instant Messaging and Web Conferencing.

10. Start the Instant Messaging and Web Conferencing server by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Domino Server**.

11. When all the Instant Messaging and Web Conferencing services start, you will see the message `Sametime Running` on the Domino console.

4.3.4 Common Domino administrative procedures

In this section, we discuss some common Domino administration tasks that you might need to occasionally do. These tasks include:

► Registering additional servers

- Registering new users
- Scheduling replication of admin4.nsf and names.nsf
- Clearing the password on an additional server's ID file
- Activating server processes for Domino
- Populating the http-hostname field in the Domino server document
- Allowing users the ability to run Java agents

Registering additional servers

To register additional servers, complete the following steps:

1. Start the administration client by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Domino Administrator**.

2. Close the Welcome window if it opens.

3. Click the **Configuration** tab

4. On the right side, open **Tools** → **Registration** → **Server**.

5. If this is the first time you have done this, complete the following steps (otherwise, skip to the next step):

    a. Click the **Server** button in the Choose a Certifier window.

    For the Registration Server, chose your first Domino server. Click **OK**.

    b. Click **Certifier ID**.

    Browse to the certifier ID (located in C:\Lotus\Domino\Data on the first server install by default).

    c. Click **OK** and enter the certifier password.

6. Click **OK** in the Certifier Recovery Information Warning window if it opens.

7. Ensure that the registration server is your first Domino server, and the certifier is the organization you created when installing that server. Click **Continue** in the Register Servers window.

8. In the Register New Server(s) window:

    a. Enter the additional server's host name (laredo in our example).

    b. Enter an ID password.

    > **Note:** In our example, we left the Password quality scale set to **Password is optional (0)**, so we could clear the password and not have to enter it every time the Domino server was restarted. For more information about the steps used to clear the password, see "Scheduling replication of admin4.nsf and names.nsf" on page 84.

c. Select the green check box.

d. Select the server and click **Register** to register this new server.

9. It is good practice to configure scheduled replication between the first server and this additional Domino server at this point to keep the important databases synchronized. For the steps to configure the scheduled replication, see "Scheduling replication of admin4.nsf and names.nsf" on page 84.

Registering new users

To register new users, complete the following steps:

1. Start the administration client by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Domino Administrator**.

2. Close the Welcome window if it opens.

3. Click the **Configuration** tab

4. On the right side, open **Tools** → **Registration** → **Person**.

5. If this is the first time you have done this, complete the following steps (otherwise, enter the certifier password and skip to the next step):

   a. Click the **Server** button in the Choose a Certifier window.

      For the Registration Server, chose your first Domino server. Click **OK**.

   b. Click **Certifier ID**.

      Browse to the certifier ID (located in C:\Lotus\Domino\Data on the first server install by default).

   c. Click **OK** and enter the certifier password.

6. Click **OK** in the Certifier Recovery Information Warning window if it opens.

7. In the Register Person -- New Entry window:

   a. Ensure that the registration server is your first Domino server.

   b. Enter the first name, last name, UID in the Short name field, and the password.

   c. Select **Password Options**:

      i. Select **Set internet password** (this is the password the user will use to log in to database over the Web, or for LDAP connections from WebSphere Portal).

      ii. Click **OK**.

   d. Select the **Advanced** options. Select the **Mail** tab and change the following values:

      i. Change your mail server if you do not want it to be the first Domino server.

ii. Change the mail file template to Domino Web Access or Mail (R6) depending on the template and portlet you want to use.

e. Select the green check box.

f. Select the user and click **Register** to register the user and create the user's mail file.

Scheduling replication of admin4.nsf and names.nsf

To schedule replication for admin4.nsf and names.nsf, the main address book database for Domino servers, complete the following steps:

1. Start the Notes client by selecting **Start** → **Programs** → **Lotus Applications** → **Lotus Notes**.

2. Open the Domino Directory from the first server:

   a. Select **File** → **Database** → **Open**.

   b. Chose the first server in the drop-down list (kingston/itso in our example)

   c. Select <your domain> directory (itso directory in our example) in the server picker list.

   d. Click **Open**.

3. Open the **Configuration** → **Servers** → **Connections** view.

4. Click the **Add Connection** button.

5. On the Basic tab, we have the following values:

   – Connection type: **Local Area Network**.

   – Source server: `kingston`/`itso` (first Domino server).

   – Destination server: `laredo`/`itso` (the newly created server).

   – Source domain: `itso`.

   – Destination domain: `itso`.

   – Use the port(s): Click **Choose ports** and select **TCPIP**.

   – Optional network address: `laredo.itsc.austin.ibm.com`.

6. On the **Replication/Routing** tab, you should have the following values:

   – Replication task: **Enabled**

   – Replicate databases of: **Low & Medium & High**

   – Files/Directory Paths to Replicate: `admin4.nsf`; `names.nsf`

7. On the Schedule tab, you should have the following values:

   – Schedule: **Enabled**.

- Connect at times: This depends on when you want replication to take place. Our example set this to 01:00 AM - 11:00PM.

- Repeat interval of: This depends on how often you want the database to replicate. In a development environment, where values change quite frequently, and there are very few users, you should set this very low (30 min). Then, move it up after you move into production.

- Days of the week: Our example set this to every day.

8. Click **Save & Close**.

Clearing the password on an additional server's ID file

To clear a password on an additional server's ID file, compete the following steps:

1. Ensure that the Domino server is shut down.

2. Open the Notes client on the administrator client.

3. Select **File → Security → Switch ID** and use the server.id file from the additional server data directory. Click **Open**, and then enter your password.

4. Select **File → Security → User Security**, and then enter your password.

5. Click the **Change Password** button, and then enter your password.

6. Click the **No Password** button.

7. Click **Yes** to confirm your choice.

8. Click **OK** in the password change success window.

9. Click **OK** to close the User Security window.

10. Select **File → Security → Switch ID** and switch back to your original Domino ID file.

11. Restart the Domino server.

You should no longer have to enter a password when it starts.

Activating server processes for Domino

There are two approaches to activating a task in the Domino server:

► To activate a task, issue the `load <taskname>` command from the Domino server.

► To enable a task to load automatically every time the Domino server starts, modify the Notes.ini file in the Domino program directory. Ours resides in the C:\Lotus\Domino directory. Modify the ServerTasks parameter and add the task name to it.

For the WebSphere Portal implementation, there are three tasks that we might need to use:

► The HTTP task: Communication between WebSphere Portal server and Domino servers is mostly performed over HTTP. This protocol must be activated on all Domino servers.

► The DIIOP task: DIIOP is used to perform database selection or database picker from a portlet.

► The LDAP task: Notes Mail and Team Workplace store some information about the registered user in the Domino directory, even when your primary identity server is not the Domino LDAP server. You need at least one LDAP Domino task running.

Populating the http-hostname field in the Domino server document

To populate the http-hostname field in the Domino server document, complete the following steps:

1. Start the Domino administrative console.

2. Open the address book from the LDAP server.

3. Select **Server** → **Servers** to navigate to the Servers view.

4. Double-click the server you want to appear in the picker, or the user's Domino mail server.

5. Go to the **Internet Protocols** tab.

6. On the **HTTP** tab, select the **Host(s) Name** option to add the host name for this server. In our example, we enter kingston.itsc.austin.ibm.com.

7. Click **Save and Close**.

Allowing users the ability to run Java agents

To allow an HTTP task to load Java agents, complete the following steps:

1. Start the Domino administrative console.

2. Open the address book from the desired server.

3. Select **Server** → **Servers** to navigate to the Servers view.

4. Double-click the server document you want to configure.

5. Make the following configuration changes to the server document:

   a. On the **Basics** tab, make sure the **Fully Qualified Internet Host Name** field contains the fully qualified name you entered in the browser to access this server.

b. Switch to the **Ports** tab. On the Notes Network Ports subtab, make sure that the top line has the port set to **TCPIP** and the Net Address set to the fully qualified name of the server. Make sure this port is set to **Enabled**.

c. Switch to the **Internet Protocols** tab. On the HTTP subtab, select **Yes** for the option Allow HTTP Clients To Browse Databases.

d. Switch to the **Security** tab. For troubleshooting and development purposes, set the following two fields to ***** in the Programmability Restrictions section:

```
Run restricted Java/JavaScript/COM: *
Run unrestricted Java/JavaScript/COM:  *
```

> **Note:** After you have this working, you might want to restrict these fields to a subset of users. You can do this, but be careful about how you enter the names.
>
> First, the Domino server you are connecting to must be included with the full canonical name (for example, kingston/itso).
>
> Next, add any users, groups, or both, that you want to receive a list of databases when placing a portlet in edit mode. You can also use an asterisk (*) as a wild card.
>
> If you want to add the user wpsadmin, you would add the following to the field:
>
> ```
> uid=wpsadmin/cn=users/o=ibm/c=us
> ```
>
> To add all members in the /o=ibm/c=us organization, add the following to the field:
>
> ```
> */o=ibm/c=us.
> ```
>
> The group cn=wpsadmins would not work in our example, because that group is not located in the Domino Directory.

4.4 Installing Domino Extended Products portlets

The Domino Extended Products portlets consist of the following portlets:

► Domino portlets:

– Lotus Domino Web Access (formerly called iNotes): Provides Welcome, Mail, Calendar, To Do List, Contacts, and Notebook functions for mail databases built with the Domino Web Access template.

– Lotus Notes View: Displays the contents of any Lotus Notes database.

- Lotus Notes Mail: Displays the complete contents of a Lotus Notes Mail database including mail, calendar, and task views.

- My Lotus Notes Mail: Displays the contents of the user's Inbox.

- My Lotus Notes Calendar: Displays the user's Calendar.

- My Lotus Notes To Do: Displays the user's To Do view.

- Lotus Notes Discussion: Displays Notes databases built with the Discussion Database template.

- Lotus Teamroom: Displays Notes databases built with the Team Room Database template.

► Lotus Team Workplace portlets:

- Lotus QuickPlace: Displays up to six different Lotus QuickPlaces in separate browser windows.

- Lotus QuickPlace Inline: Displays a Lotus QuickPlace inside the portlet.

- My Team Workplaces: Enables users to find, work in, and request new team workplaces, as well as view workplace details.

► Lotus Instant Messaging and Web Conferencing portlets:

- Lotus Sametime Connect: Launches the Sametime instant messaging applet in a separate browser window.

- Sametime Contact List: Displays a Lotus instant messaging list.

- Sametime Who Is Here: Displays a Lotus Sametime list of users of the portal page.

- Lotus Web Conferencing: Enables users to find, attend, and schedule e-meetings, as well as view meeting details.

- People Finder portlet: Enables users to implement both quick search and advanced search for locating people and information about people.

For a more detailed description of how each portlet works, refer to the *IBM WebSphere Portal Information Center*, available at:

```
http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wps/collabportlet.html
```

The WebSphere Portal interaction with the back-end Domino Extended Products is performed using the Collaborative Services. The Collaborative Services are a collection of Java classes stored in the cs.jar file. This file is located in the <wp_root>\shared\app directory and configured using the CSEnvironment.properties located in the <wp_root>\shared\app\config directory.

Running any one of the **wpsconfig** batch commands for configuring Lotus Domino, Team Workplace or Instant Messaging and Web Conferencing will

install these files to the WebSphere Portal server. We discuss additional changes to the CSenvironment.properties file later.

To install the portlets, we need to configure WebSphere Portal and then install the portlets. We describe this procedure in the following sections. Further configuration of the portlets is also necessary. You will only need to follow the steps for the portlets you want to use.

4.4.1 Configuring WebSphere Portal for collaborative portlets

The Domino LDAP server needs to be configured to enable some functionality of the portlets. See 4.3.4, "Common Domino administrative procedures" on page 81. After you verify that the Domino LDAP server is working correctly, complete the following steps to modify the wpconfig.properties file for collaborative portlets:

1. Stop the WebSphere_Portal application server using the following command:

   ```
   stopServer WebSphere_Portal -user wpsbind - password wpsbind
   ```

2. Locate the <wp_root>/config/wpconfig.properties file. Copy the file to create a backup before changing any values.

3. There are several changes that you might want to perform in this file. Use a text editor to open the <wp_root>/config/wpconfig.properties file.

   – The modification to enable the Domino LDAP directory is shown in Example 4-2.

Example 4-2   Domino Directory section of the wpconfig.properties file

```
# Description: Lotus Collaborative Components required properties
#              to enable Lotus Domino Directory

# LCC.DominoDirectory.Enabled: Is Lotus Domino Directory enabled in the
# environment?
# { true | false }
LCC.DominoDirectory.Enabled=true

# LCC.DominoDirectory.Server: The Lotus Domino Directory server name.
# { hostname | ip address }
LCC.DominoDirectory.Server=toronto.itsc.austin.ibm.com

# LCC.DominoDirectory.Port: The port number for the Lotus Domino Directory
# server.
# { port number }
LCC.DominoDirectory.Port=389

# LCC.DominoDirectory.SSL: Is SSL used to connect to the Lotus Domino Directory
# Server?
# { true | false }
LCC.DominoDirectory.SSL=false
```

– The changes for the QuickPlace server connection is shown in
    Example 4-3.

Example 4-3   QuickPlace section of the wpconfig.properties file

```
# Description: Lotus Collaborative Components required properties
#              to enable Lotus QuickPlace

# LCC.QuickPlace.Enabled: Is Lotus QuickPlace enabled in the environment?
# { true | false }
LCC.QuickPlace.Enabled=true

# LCC.QuickPlace.Server: The Lotus QuickPlace server name.
# { hostname | ip address }
LCC.QuickPlace.Server=kingston.itsc.austin.ibm.com

# LCC.QuickPlace.Protocol: The protocol used to connect to the Lotus QuickPlace
# server.
# { http | https }
LCC.QuickPlace.Protocol=http

# LCC.QuickPlace.Port: The port number for the Lotus QuickPlace server.
# { port number }
LCC.QuickPlace.Port=80
```

  – The changes for the Sametime server connection is shown in
    Example 4-4.

Example 4-4   Sametime section of the wpconfig.properties file

```
# Description: Lotus Collaborative Components required properties
#              to enable Lotus Sametime

# LCC.Sametime.Enabled: Is Lotus Sametime enabled in the environment?
# { true | false }
LCC.Sametime.Enabled=true

# LCC.Sametime.Server: The Lotus Sametime server name.
# { hostname | ip address }
LCC.Sametime.Server=laredo.itsc.austin.ibm.com

# LCC.Sametime.Protocol: The protocol used to connect to the Lotus Sametime
# server.
# { http | https }
LCC.Sametime.Protocol=http

# LCC.Sametime.Port: The port number for the Lotus Sametime server.
# { port number }
LCC.Sametime.Port=80
```

4. Save the file.

5. Run the following commands:

```
WPSconfig.bat lcc-configure-dominodirectory
WPSconfig.bat lcc-configure-quickplace
WPSconfig.bat lcc-configure-sametime
```

6. Restart the WebSphere_Portal application server using the following command:

```
startServer WebSphere_Portal
```

4.4.2 Installing the Domino Extended Products portlets

The Lotus Domino Extended Products portlets are grouped into two separate installation processes:

▶ The collaborative portlets, where you can choose specific portlets to be installed, consisting of:

– Lotus Discovery Server™ (not discussed in this book)
– Domino Web Access (formerly called iNotes)
– Notes and Domino
– Lotus QuickPlace
– Inline QuickPlace
– Lotus Sametime Connect
– Domino.Doc® (not discussed in this book)
– Quick e-Mail (not discussed in this book)
– Quick Appointment (not discussed in this book)
– Web Page (not discussed in this book)

▶ The Collaboration Center portlets, where you have to install all portlets, consisting of:

– Sametime Contact List
– Sametime Who Is Here
– People Finder
– Lotus My Team Workplaces (formerly called QuickPlace)
– Lotus Web Conferencing (formerly called Sametime)

Installing the collaborative portlets

Whether you are installing every portlet or just a select few, first ensure that the WebSphere Portal server is up and running. The command syntax to install the collaborative portlets is:

```
xmlaccess -in ..\config\work\<xml_file> -user admin_name -pwd admin_password
-url server_name:port_number/portal_server_context_root/config
```

Where:

`xml_file`	The XML property file.
`admin_name`	Administrator name.
`admin_password`	Administrator password.
`server_name`	Server TCP/IP host name of WebSphere Portal.
`port_number`	Port number of the WebSphere Portal server.
`portal_server_context_root`	WebSphere Portal server context. The default is wps.

The command that we used to install all the collaborative portlets uses lccportlets.xml. The complete command is:

```
xmlaccess -in ..\config\work\lccportlets.xml -user wpsadmin -pwd wpsadmin -url
pretoria.itsc.austin.ibm.com:9080/wps/config
```

To install a single portlet, replace the XML file in the above command with the corresponding XML file for the portlet you want from the following list:

`lcc_domdoc_portlet.xml`	Domino.Doc
`lcc_inotes_portlet.xml`	Domino Web Access portlet
`lcc_lotusds_portlet.xml`	Lotus Discovery Server Knowledge Map, Lotus Discovery Server Mini-Search Results, Lotus Discovery Server Search Results
`lcc_notes_portlet.xml`	Notes and Domino
`lcc_quickappointment_portlet.xml`	Quick Appointment
`lcc_quickemail_portlet.xml`	Quick e-Mail
`lcc_quickplace_portlet.xml`	Lotus QuickPlace
`lcc_quickplace2_portlet.xml`	Inline QuickPlace
`lcc_sametime_portlet.xml`	Lotus Sametime Connect
`lcc_webpage_portlet.xml`	Web Page

Installing the Collaboration Center portlets

The Collaboration Center portlets are shipped with the WebSphere Portal V5.0.2 in CD 12.

> **Important:** The WebSphere Portal Version 5.0.2 CDs come with two versions of the Collaboration Center. CD 8 ships the Collaboration Center Version 5.0. This version will not work correctly with WebSphere Portal V5.0.2. Instead, you need the Collaboration Center files from CD 12.

Installing the Collaboration Center is a two-step process:

1. Install the Collaboration Center. From CD 12, run the following command from a command prompt:

   ```
   install "C:\WebSphere\PortalServer"
   ```

   This command moves the files necessary to deploy the Collaboration Center portlets onto the WebSphere Portal server. After this command runs, you should see the following message:

   ```
   Collaboration Center install success
   ```

2. Deploy the Collaboration Center portlets into WebSphere Portal. From the directory <wp_root>/config, run the following command:

   ```
   WPSconfig.bat cc-deploy-portlets
   ```

Now all the Domino Extended Products portlets are installed into WebSphere Portal. Next, you need to configure the Domino Extended Products to work with these portlets before users can use them.

4.4.3 Configuring the Collaboration Services to bind to Domino LDAP

To configure the Lotus Collaboration Services to bind to the Domino LDAP, complete the following steps:

1. Edit the CSEnvironment.properties file on your WebSphere Portal server. It should be located in the \WebSphere\PortalServer\shared\app\config.

2. Uncomment and enter the following entries:

   – `CS_SERVER_DOMINO_DIRECTORY_1.userid`

     The user ID in CS_SERVER_DOMINO_DIRECTORY_1.userid must have at least reader access to the names.nsf in the Domino LDAP server. We use the following Domino LDAP canonical name: `cn=domino admin,o=itso`.

   – `CS_SERVER_DOMINO_DIRECTORY_1.encryptedpwd`

     The encrypted password for CS_SERVER_DOMINO_DIRECTORY_1.encryptedpwd can be acquired using the **PropFilePasswordEncoder** command from WebSphere\AppServer\bin directory. As an example, to encrypt a password called pwd0dwp, create a text file (pass.txt) with the content shown in Example 4-5 and run the **PropFilePasswordEncoder pass.txt password** command. The pass.txt file will contain the encrypted password.

Example 4-5   Sample password file

```
password=pwd0dwp
```

3. Save CSEnvironment.properties and restart WebSphere Portal for the change to take effect.

Now, Domino server names should appear in the server picker, and users should be able to use the automatically detect mail database for the Domino Web Access and Notes Mail portlets. If you continue to experience problems with the picker or auto detection of mail database, see the following troubleshooting Technotes:

► *Troubleshooting Pickers in Lotus Collaborative Portlets*, Technote 1157249:

   http://www.ibm.com/support/docview.wss?rs=0&uid=swg21157249

► *Troubleshooting Automatic Detection of your Mail File with the Different Lotus Collaborative Portlets*, Technote 1157029:

   http://www.ibm.com/support/docview.wss?rs=0&uid=swg21157029

4.4.4 Enabling server access for portlets

After the collaboration portlets has been installed, there are still several configuration tasks that need to be performed to finalize the setup. The following list describes these tasks:

► Enabling single sign-on (SSO). This can be performed by exchanging the LTPA key in the LTPA domain. SSO with LTPA can be used for WebSphere Portal with the Domino server or WebSphere Portal and the Sametime server. We discuss the basic LTPA SSO in 4.4.5, "Configuring single sign-on" on page 95.

► Setting the collaboration service to bind to the Domino LDAP for the QuickPlace Inline portlet and Notes Web access. This is performed after the Domino server has been populated.

► Configuring QuickPlace:

   – domcfg.nsf
   – Update Notes.ini
   – Configure My Team Workplace (Qpservlet and search)

► Configuring Sametime:

   – Instant Messaging portlet
   – Web conferencing portlet
   – SSO Qp and ST
   – Sametime awareness
   – Sametime meeting

► Configuring People Finder.

► Placing a portlet in a page for access.

4.4.5 Configuring single sign-on

The type of single sign-on (SSO) that we discuss in this section is based on the Lightweight Third Party Authentication (LTPA) token mechanism that is generated by WebSphere Portal to access back-end servers that share an authentication directory. In Chapter 6, "Incorporating IBM Tivoli Access Manager for e-business" on page 167, we discuss SSO based on the LTPA token that is generated by the reverse proxy (WebSEAL) and we also discuss Trust Association Interceptor (TAI) as an SSO mechanism.

For additional information, see the Technote *Troubleshooting WebSphere Portal, Sametime, QuickPlace and Domino SSO Issues*, Technote 1158269, available at:

```
http://www.ibm.com/support/docview.wss?rs=0&uid=swg21158269
```

Checking Web SSO configuration in Domino

The SSO definition in Domino is stored in the names.nsf database as the Web SSO configuration document. First, you might want to check whether a document already exists:

1. Using a Notes client, open names.nsf on the Domino server with which you are working.

2. Select **Configuration** → **Web** → **Web Configurations** to open the Web Configurations view.

3. If you see a *-Web SSO Configurations- twistie with a Web SSO Configuration for LTPA document, as shown in Figure 4-2 on page 96, the Web SSO configuration document already exists.
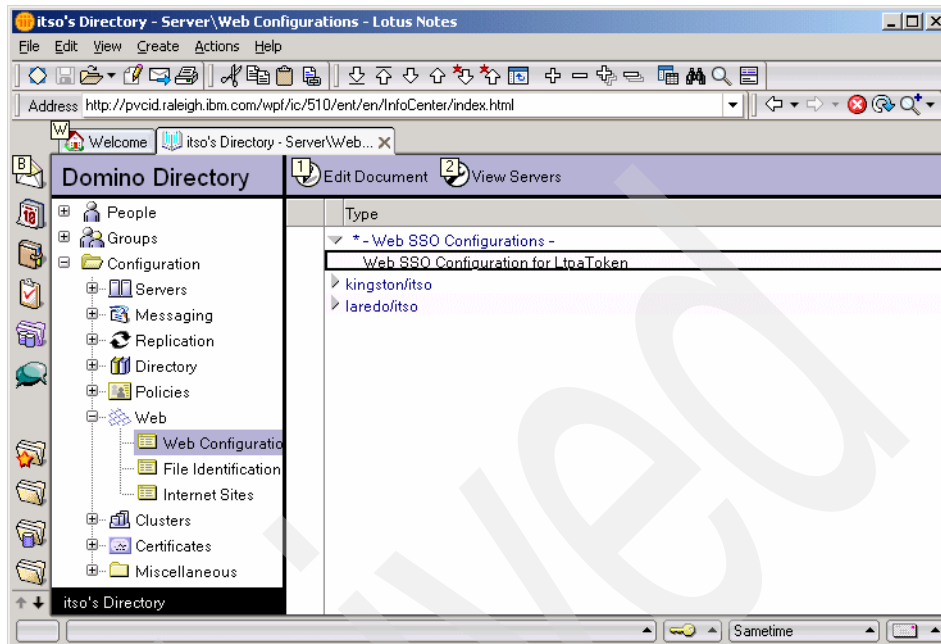
Figure 4-2 Web Configurations view

You can do one of the following:

- ► Use the Web SSO configuration document, if you are certain that it contains the appropriate WebSphere LTPA key. All you need to do is add the Domino servers in the Domino Server Names field and replicate the change using the `rep` command. For example, to replicate for our kingston machine, use the command `rep kingston/Itso names.nsf`.

- ► If you are not sure of the Web SSO configuration, you can delete the document and simply follow the following procedure to recreate a Web SSO configuration.

Configuring single sign-on for Domino

To configure single sign-on between WebSphere Portal and the Domino mail and application servers, complete the following steps:

1. Create the WebSphere LTPA key:

   a. Start the Portal WebSphere Administration Console and log in and log in as wpsbind. In our example, we use the following URL:

      `http://pretoria.itsc.austin.ibm.com:9081/admin`

   b. Select **Security** → **Authentication Mechanisms** → **LTPA**.

c. Type a password and provide a name path and file name for the key file.

> **Tip:** Remember this password, because you must enter it when you import the LTPA key into the Domino server and when you create LTPA junctions in Tivoli Access Manager.

d. Click the **Export Keys** button.

e. Click **Save** to apply the changes to the master configuration.

f. Click **Save** in the next window.

g. Log out of the WebSphere Administration Console.

h. Copy the key file that you created to a location that is accessible by the Domino server.

2. Import the key into Domino:

a. Start the Domino administrative client and open the address book (names.nsf) for the server.

b. Change to the **Server** → **Servers** view.

c. Click the **Web** button, and then select **Create Web SSO Configuration**.

d. Enter the domain suffix in the **Token Domain** field. This should match the domain name you entered in the WebSphere Portal server.

e. Add the Domino hierarchical name of the Domino servers that will participate in the SSO domain in the **Domino Server Names** field. You do not need to enter the names of the WebSphere Application Server.

f. Select **Keys** → **Import WebSphere LTPA keys**.

g. Enter the path and name of LTPA key file, and click **OK**.

h. Enter the password for the LTPA key and click **OK**.

i. Click **OK** in the message window that states that the key import is successful.

j. Click the **Basics** tab and add a \ (backward slash) to the **LDAP Realm** field. For our IBM Directory server host, we used:

    phoenix.itsc.austin.ibm.com\:389

k. Click **Save and Close**.

3. Enable multiserver single sign-on authentication. This enables Domino to look for LTPA tokens in the browser to authenticate users.

a. Open the server document of the Domino server.

b. Click the **Internet Protocols** tab, and then the **Domino Web Engine** tab.

c. Next to Session authentication, select **Multi-server**.

d. Click **Save and close**.

e. Exit the Domino administrative client.

f. You might want to replicate the new document to each servers specified in the Domino Server Names field using the `rep` command.

g. Restart all the Domino servers.

4. Synchronize the IBM Directory server and Domino server's user registry.

The user registries will need to be synchronized together on every Domino server in your environment, including both the Domino LDAP and mail and application servers. You should complete these steps on the Domino LDAP server first, and then replicate the changes out to the mail and application servers.

a. Start the Domino administrative console.

b. Open the address book from the LDAP server.

c. Change to the Person view.

d. Open the Person document for a user with whom you are attempting to enable SSO.

e. Add the user's IBM Directory Server DN (replacing the comma with a forward slash) and login name to WebSphere Portal in the **User name** or **Short name** field. In our example, we create a user in IBM Directory Server with the DN uid=iuser1,cn=user,o=ibm,c=us, and with that user logged into WebSphere Portal as iuser1. Next, we created a user in Domino called iNotes User1/ITSO. The user iNotes user1's modified Person document from the example is shown in Figure 4-3 on page 99.

> **Important:** In the example, the IBM Directory Server DN is added as the last line of the User name field, and the Portal login ID (UID) is added to the Short name field. You can add these values anywhere you like in either field, as long as they are not the top line of the User name field. This must remain the Domino canonical name.

f. Ensure that the password in IBM Directory Server and the Internet password in the Person document match.

Figure 4-3   Person document example

Additional configuration for Lotus Team Workplace

For the Lotus Team Workplace server, perform the following additional steps that to enable single sign-on:

1. Create the Domino Web Server Configuration database, domcfg.nsf:

    a. From a Notes client, select **File** → **Database** → **New**.

    b. We use the following properties:

        • Server: `kingston/Itso` (Team Workplace server)
        • Title: `Web Server Configuration`
        • File name: `domcfg.nsf`
        • Template: **Domino Web Server Configuration (6)** (domcfg5.ntf); this template is shown with the Advanced templates.

    c. Click **OK**.

    d. Open the newly created Web Server Configuration database.

    e. Click **Add Mapping**.

    f. In the Mapping document, fill in the following:

        • Applies to: **All Web Sites/Entire Server** (you can also restrict SSO to specific virtual servers.)
        • Target Database: `quickplace/resources.nsf`
        • TargetForm: `QuickPlaceLoginForm`

g. Click **Save & Close**.

2. Update the Notes.ini file:

   a. Open the Notes.ini file in the \Lotus\Domino directory of your Team Workplace server in a text editor.

   b. Add the directive `NoWebFileSystemACLs=1` to the file. Do not place this as the last line of the file.

3. Restart the Domino server for the changes to take effect.

Testing single sign-on

Perform the following steps to test single sign-on between WebSphere Portal and your Domino mail or application server.

1. Sign on to WebSphere Portal.

2. Depending on the application that you are testing, you can perform one of the following:

   – For a Domino mail application, access a Domino database that the access control list (ACL) has the Default and Anonymous access set to No Access. In our example, we tested with iNotes user1's mail file, and no user name and password prompt should appear:

      `http://kingston.istc.austin.ibm.com/mail/iuser1.nsf`

   – For a Lotus Instant Messaging and Web Conferencing server, open stcenter.nsf and click **Attend a Meeting**. Your name should appear. Our example uses the URL:

      `http://laredo.itsc.austin.ibm.com/stcenter.nsf`

   – For a Lotus Team Workplace server, point to the main QuickPlace page and your name should appear in the top left corner. Our QuickPlace URL is:

      `http://kingston.itsc.austin.ibm.com/quickplace`

4.4.6 Lotus Team Workplace portlets settings

Perform the following steps for each Lotus Team Workplace portlet:

► For the Lotus QuickPlace portlet, you only need to configure single sign-on between WebSphere Portal and QuickPlace.

► For the Lotus QuickPlace Inline portlet, you need to complete the following steps:

   – Configure single sign-on between WebSphere Portal and QuickPlace.

   – Configure Domino LDAP for QuickPlace Inline.

– Optionally, for the database picker, you can configure http-hostname and bind ID for the Domino LDAP and the DIIOP task.

► For the My Lotus Team Workplaces portlet, you need to configure single sign-on between WebSphere Portal and QuickPlace and configure the My Team Workplace portlet.

4.4.7 Configuring the My Team Workplace portlet

After installing the My Team Workplace portlet, you should update the portlet with the host name of your Lotus Team Workplace server. To update the My Team Workplace portlet with the host name, complete the following steps:

1. Sign in to WebSphere Portal as the Portal administrator, wpsadmin in our example.

2. Click **Administration**.

3. Select **Portlets** → **Manage Portlets**. Navigate to and select the My Lotus Team Workplaces portlet and click **Modify Parameters**.

4. In the **QuickPlaceHostName** field, type the fully qualified host name of the Team Workplace server. We use kingston.itsc.austin.ibm.com.

5. Click **Save** to save your changes.

The My Team Workplace portlet will now attempt to communicate with your Team Workplace server. However, the portlet will be unable to do so until you complete the configuration steps in the following two sections.

Configuring qpservlet

The qpservlet is a small Java program that must run on the Lotus Team Workplace server so that WebSphere Portal can communicate with the Team Workplace server to provide all the functionality in the My Team Workplace portlet.To set up and test the qpservlet, perform the following steps:

1. Create the servlets.properties file.

   Create a file named servlets.properties in the Domino data directory.

   This file should contain the following:

   ```
   servlet.QPServlet.code=com.lotus.cs.util.QPServlet
   ```

   There needs to be a hard carriage return at the end of this line; be sure to press the Enter key at the end of the line.

2. Install the Lotus Collaborative Components files:

   a. Copy the cs.jar and People.tld files from the WebSphere Portal server to a directory on your Team Workplace server. These files need to reside in the same directory. In our example, we copied these files to the Domino program directory C:\Lotus\Domino.

      - The cs.jar file is located in the \WebSphere\PortalServer\shared\app directory.
      - The people.tld file is located in the \WebSphere\PortalServer\shared\app\WEB-INF\tld directory.

   b. Edit the JavaUserClassesExt line in your Notes.ini file to include the full path of the cs.jar file. The JavaUserClassesExt line from our example is shown in Example 4-6.

Example 4-6   JavaUserClassesExt from Notes.ini

```
JavaUserClassesExt=QPJC1,QPJC2,QPJC3,QPJC4,QPJC5,QPJC6
QPJC1=C:\LOTUS\DOMINO\quickplace.jar
QPJC2=C:\LOTUS\DOMINO\xercesImpl.jar
QPJC3=C:\LOTUS\DOMINO\xalan.jar
QPJC4=C:\LOTUS\DOMINO\xml-apis.jar
QPJC5=C:\LOTUS\DOMINO\log4j-118compat.jar
QPJC6=C:\LOTUS\DOMINO\cs.jar
```

3. Enable the Domino servlet manager:

   a. From Notes client, open the Domino Directory (Names.nsf) on the Team Workplace server.

   b. Open the Team Workplace server document.

   c. Select **Internet Protocols** → **Domino Web Engine**, and set the Java Servlet Manager field to **Domino Servlet Manager**.

   d. From the **Security** tab:

      - Set Run Restricted Java/JavaScript/COM: to *
      - Set Run Unrestricted Java/JavaScript/COM: to *

      > **Note:** *After* you have this working, you might want to restrict these fields to a subset of users. You need to be careful about how you enter the names:
      >
      > ► The Domino server you are connecting to must be included with the full canonical name (for example, kingston/itso).
      > ► Add any users and groups who will use the My Team Workplaces portlet. You can also use an asterisk (*) as a wild card. You need to add the users that are known by the Domino Directory.

e. Click **Save and Close**, and exit out of the Notes client.

4. Create a Servlet subdirectory in the \Data\Domino directory with nothing in it. In our example, we created the directory structure C:\Lotus\Domino\Data\Domino\Servlet.

5. Anytime you correct or change any of these settings, you will need to restart the Domino server for the change to take effect.

After restarting the Team Workplace server, make sure that the qpservlet loads correctly by addressing the servlet URL. Our servlet URL is:

```
http://kingston.itsc.austin.ibm.com/servlet/QPServlet?actionType=69
```

The browser should return a message similar to the following:

```
QPServlet:LCS Build [Version][Date][WPS buildstream]=[KS3224wa][0302.2600][5.0]
Posted Build
Use with QP3.0.1
```

For more information about troubleshooting qpservlet, refer to the Technote *Error: "Connection to QuickPlace Server Could not Be Established" in My Team Workplace Portlet*, 1159319, available at:

```
http://www.ibm.com/support/docview.wss?rs=0&uid=swg21159319
```

Configuring Search All Places

Four features of the My Team Workplaces portlet, My Tasks, My Pages, Search This Place, and Search My Workplaces, use a feature of the Team Workplaces server that is not configured by default. This feature is Search All Places. The information included here is based on the instructions in the *Lotus Team Workplace Administrator's Guide*, G210-1656, available at:

```
http://www.lotus.com/ldd/notesua.nsf/find/quickplace
```

The *Administrator's Guide* recommends creating the place catalog server as its Team Workplace. To set up the Search All Places feature, complete these steps:

1. Configure Domain Search and Domain Indexer:

   a. Open the Domino Directory (Names.nsf) from the Team Workplace server in a Notes client.

   b. Select **Configuration** → **Servers** → **All Server Documents** and open the Team Workplace server document.

   c. Select the **Server Tasks** → **Domain Catalog** tab.

   d. Select **Enabled** in the Domain Catalog field. This step starts the Catalog task and creates the Domain Catalog. You run the Catalog task to keep the Database Catalog up to date. You might do this on a schedule, for example, by including the task in the Notes.ini setting, ServerTasksAt1.

e. Click the **Domain Indexer** tab.

f. Click **Enabled** in the Schedule field to enable the Domain Indexer task. Specify a schedule for running the Domain Indexer.

   For more information about setting up Domino Domain Search, see the following topics in *Domino Administration Help*: "Enabling Domain Search," "The Database Catalog," and "The Domain Search Index." The *Domino Administrator Help* is available at:

   http://www.lotus.com/ldd/notesua.nsf/find/domino

2. Configure the Search Places settings in the qpconfig.xml file:

a. Open the qpconfig.xml file using a text editor.

b. Scroll down to the Search Places section and remove the following lines from the beginning and end of Search Places section, respectively:

   ```
   <!-- =============== START OF SAMPLE =================
   =============== END OF SAMPLE ================== -->
   ```

c. Modify the Search Places tags for your environment. Example 4-7 shows our configuration.

Example 4-7   Search Places section of the qpconfig.xml file

```
<search_places enabled="true" log_level="0" anonymous="true">
    <domain_catalog_server ssl="false">
          <port>80</port>
          <domino_server_name>kingston/itso</domino_server_name>
          <path_prefix></path_prefix>
          <hostname>kingston.itsc.austin.ibm.com</hostname>
    </domain_catalog_server>
</search_places>
```

d. Restart the Domino server for these setting to take effect.

Test the Search All Places feature in Team Workplace:

1. Sign in to a place you have created.

2. Click **Search**.

   You should see three options:

   – Search All Places
   – Search This Place
   – Folder

3. Select the **Search All Places** option and search something you know will return results, for example, Welcome.

If you do not see the Search All Places option or the search for Welcome does not return results, see the Technote *Troubleshooting Script for Setting QuickPlace to Search Across All Places*, 1106449, for further assistance:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21106449

4.4.8 Lotus Instant Messaging and Web Conferencing portlets

There are four portlets designed to work with your Lotus Instant Messaging and Web Conferencing server.

► For the Sametime Connect and Who Is Here portlet, you need to configure single sign-on between WebSphere Portal and Sametime.

► For the Sametime Contact List portlet, you need to complete the following:

– Configure single sign-on between WebSphere Portal and Sametime.
– Allow the WebSphere Portal Instant Messaging server application access to the Instant Messaging and Web Conferencing server (Contact List portlet only).

► For the My Lotus Web Conferencing portlet, you need to complete the following:

– Configure single sign-on between WebSphere Portal and Sametime.
– Configure the Lotus Web Conferencing portlet.

► Other portlets with the awareness capabilities of the Instant Messaging server show the users' messaging status with a green, yellow, or do not disturb dot message next to their names. These portlets are the Notes and Domino, People Finder, Lotus Web Conferencing, and My Lotus Team Workplaces portlets. For these portlets to show awareness, you will need to configure single sign-on between WebSphere Portal and Sametime.

4.4.9 Allowing Contact List portlet to access Instant Messaging server

The Lotus Contact List portlet connects to the Instant Messaging server to retrieve a user's buddy list. You need to configure the Instant Messaging server to allow the application on the WebSphere Portal server to connect to the Instant Messaging server. The following steps explain how to do this:

1. Open the sametime.ini file in a text editor off the Instant Messaging server. This file is located in the Domino program directory (C:\Lotus\Domino in our example).

2. Configure Sametime to accept all IP addresses as trusted. To do this, add the following line to the Debug section at the end of the file:

```
[Debug]
VPS_BYPASS_TRUSTED_IPS=1
```

> **Note:** In a production environment, you can add the IP address of the WebSphere Portal server machine to the list of IP addresses of trusted servers and remove the [Debug] section so that you are not accepting all IP addresses as trusted. To do this, add the following line to the Configuration section:
>
> ```
> [Config]
> VPS_TRUSTED_IPS=trusted IP address, trusted IP address
> ```
>
> Where *trusted IP address* is the IP address of your WebSphere Portal server.
>
> If you need to add multiple trusted servers, separate your new entry from the previous entry using a comma. Do not enter the host name. Enter the IP address.

3. Restart the Instant Messaging server. After it is up and running, restart the WebSphere Portal server for the change to take effect.

4.4.10 Configuring the Lotus Web Conferencing portlet

The following additional configuration steps are specific to the Web Conferencing portlet. You need to point the portlet to the your Web Conferencing server and tell the portlet how it can authenticate with the Web Conferencing server to create and search for meetings. To configure the Web Conferencing portlet:

1. Sign in to WebSphere Portal as the Portal administrator, wpsadmin in our example.

2. Click **Administration**.

3. Select **Portlets** → **Manage Portlets**. Navigate to and select the Lotus Web Conferencing portlet and click **Modify Parameters**. Fill in the values for SametimeServerName1, SametimeUserName1, and SametimePassword1. The user that you use must be:

   – An administrator of Sametime

   – Exist only in the Domino Directory of Sametime

4. Click **Save** to save the changes, and then **Cancel** to get back to the Modify Parameters page.

If you continue to experience problems creating or searching for meetings with the Web Conferencing portlet, see the Technote *Password Errors When Using Web Conferencing Collaboration Center Portlet*, 1170825, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21170825

4.4.11 Lotus Team Workplace and Instant Messaging

If you installed and configured both Lotus Team Workplace and Lotus Instant Messaging and Web Conferencing with WebSphere Portal, you can configure Team Workplace to work with the Instant Messaging and Web Conferencing server. There are two ways the Team Workplace server can work with Instant Messaging and Web Conferencing:

- ► Sametime awareness for Team Workplace
- ► Sametime meeting for Team Workplace

Before setting up these features, ensure that SSO is working between these servers and with the WebSphere Portal server.

4.4.12 Configuring People Finder

Depending on the configuration of your LDAP directory, when People Finder is placed on a page, it might display an error. To resolve this error, complete the following steps:

1. Sign in to WebSphere Portal as the Portal administrator (wpsadmin in our example).

2. Click the configure (wrench) icon.

3. Here, you will see a list of fields in red that are causing the problem. In our example, the only field causing the problem was ibm-personalTitle, as shown in Figure 4-4 on page 108.
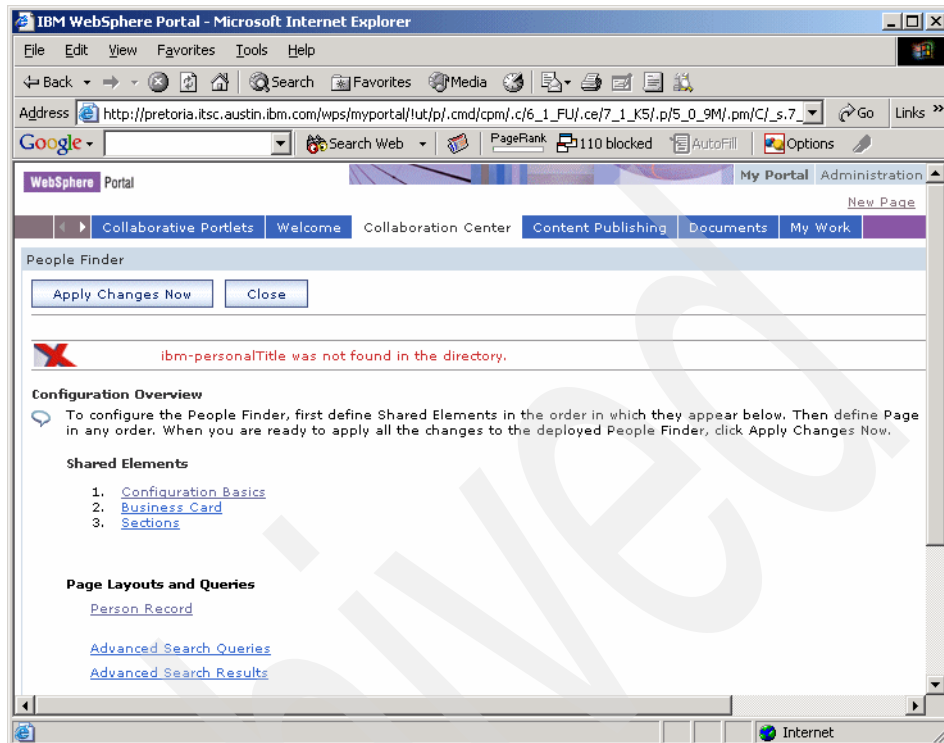
Figure 4-4   Field not found in directory

4. You need to look in every section of the portlet for references to this field and remove them. As soon as all the references have been removed, the error will be resolved. In our example, the field appeared in the Person Record section, as shown in Figure 4-5 on page 109.

Figure 4-5   Problem field in Person Record section

5. Click the trash can icon to remove the field from this section, and the field will be removed if this is the only place it appears.

6. If it disappears, click **OK** to save the changes in that section, and then **Apply Changes Now** to get back to the People Finder window. At this point, the portlet should give you search options to find people.

7. If all the fields do not disappear, continue to go through each section until they are gone. Also, many of the links, including the Person Record, contain multiple sections under the **Select a Section** title. If you remove a field from one section, click **OK**, and then **Apply Changes Now**. Finally, click the configuration (wrench) icon again to explore the additional section in each link.

Show Person Record in menu

The Show Person Record menu item, as shown in Figure 4-6, does not appear by default in the People Finder portlet for all users; it only appears for the Portal administrator.
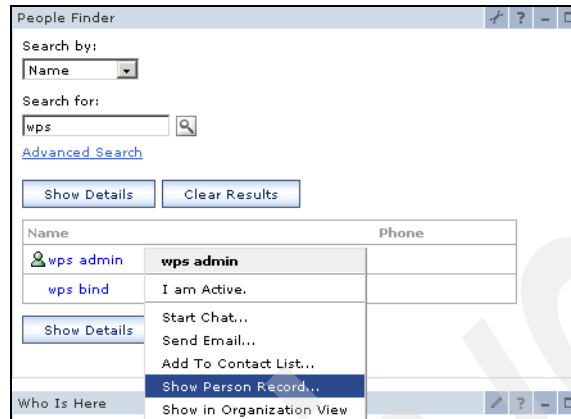

Figure 4-6   Show Person Record menu item

To enable this to appear for all users, you must give all users access to the hidden page for the People Finder. Complete the following steps:

1. Sign into WebSphere Portal as the Portal administrator.

2. Select **Administration** → **Portal User Interface** → **Manage Pages**.

3. Under Context Root, find the page lotus.workplace.hidden.page.PeopleFinder.

4. Give all authenticated portal users a user access to this page. The users will not see any additional pages in WebSphere Portal, but should now see the Show Person Record menu option.

If you continue to experience problems with the Show Person Record option, see the Technote *People Finder Portlet Does not Display "Show Person Record" Menu Choice*, 1174638, to continue troubleshooting, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174638

4.4.13 Setting up Sametime awareness and chat

To enable online awareness and chat for Team Workplace users, complete the following steps:

1. Copy the Java files required for Sametime chat and online awareness.

2. Specify the Sametime Community server for Team Workplace to use.

Copying Java files required for chat and online awareness

To copy the Java files required for chat and online awareness, complete the following steps:

1. Install the Sametime Java Toolkit:

   a. Download the Lotus Sametime 6.5.1 Java Toolkit from the following URL:

      `http://www.lotus.com/ldd/down.nsf`

   b. Extract the downloaded file into the directory <domino data>\domino\html\sametime\toolkits\st651javatk (C:\Lotus\Domino\Data\domino\html\sametime\toolkits\st651javatk in our example).

2. In the Domino data directory of the Sametime server, create the subdirectory <domino data>\Domino\html\QuickPlace\peopleonline (C:\Lotus\Domino\Data\domino\html\QuickPlace\peopleonline in our example).

3. Copy the STComm.jar, CommRes.jar, and PeopleOnline31.jar files to the QuickPlace\peopleonline subdirectory you created in the previous step. These files can be found in the following locations:

   – Files from the Instant Messaging and Web Conferencing server:

      STComm.jar and CommRes.jar: <domino data>\domino\html\sametime\toolkits\st651javatk \bin (C:\Lotus\Domino\Data\domino\html\sametime\toolkits\st651javatk\bin in our example).

   – Files from the Team Workplace server:

      PeopleOnline31.jar: <Domino data>\QuickPlace (C:\Lotus\Domino\Data\QuickPlace in our example).

Specifying the Instant Messaging server in Team Workplace

To specify the Lotus Instant Messaging server in Lotus Team Workplace, complete the following steps:

1. In a browser, type the URL of the Team Workplace server administration console (http://kingston.itsc.austin.ibm.com/quickplace in our example).

2. Click **Sign In** and sign in as a Team Workplace server administrator (qpadmin in our example).

3. Click **Server Settings** in the table of contents.

4. Click **Other Options** in the table of contents.

5. Click **Edit Options**.

6. Under the Sametime Servers heading, make sure that the Sametime Instant Messaging server is in the community field. Use the full name of the server (http://laredo.itsc.austin.ibm.com in our example).

7. Click **Next**, and then sign out of Team Workplace.

> **Note:** The Team Workplace server is not immediately integrated with Instant Messaging. Wait a few minutes for the setting to take effect, or restart the Team Workplace server to integrate it.

Testing online awareness

To test online awareness, complete the following steps:

1. In a browser, type the URL of the Team Workplace server administration console (http://kingston.itsc.austin.ibm.com/quickplace in our example).

2. Click **Sign In** and sign in as a user from the LDAP directory (wpsadmin in our example). You must log in as an external user. Sametime features are not available to local users such as qpadmin.

3. Verify that awareness is working by checking for the Awareness icon next to the name you typed when you logged in.

If a gray dot appears, but never turns green, see the Technote *Knowledge Collection: QuickPlace Issues Related to Sametime*, 1115409, for further assistance, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21115409

4.4.14 Setting up Web Conferencing meetings

To enable online meetings for Team Workplace users, complete the following steps:

1. Copy the Java files required for online meetings.

2. Specify the Web Conferencing authentication name.

3. Specify the Sametime Community server for Team Workplace to use.

Copying the Java files required for online meetings

To copy the Java files, complete the following steps:

1. Copy the STMtgManagement.jar, STCore.jar, and ibmjsse.jar files from the Domino program directory of the Sametime server (C:\Lotus\Domino in our example) to the Domino Program directory on the Team Workplace server (C:\Lotus\Domino in our example).

2. Open Notes.ini from the Domino program directory on the Team Workplace server.

3. Modify the Notes.ini setting JavaUserClassesExt to add the STMtgManagement.jar, STCore.jar, and ibmjsse.jar files, as shown in Example 4-8.

Example 4-8   Notes.ini JavaUserClassesExt section

```
JavaUserClassesExt=QPJC1,QPJC2,QPJC3,QPJC4,QPJC5,QPJC6,QPJC7,QPJC8,QPJC9
QPJC1=C:\LOTUS\DOMINO\quickplace.jar
QPJC2=C:\LOTUS\DOMINO\xercesImpl.jar
QPJC3=C:\LOTUS\DOMINO\xalan.jar
QPJC4=C:\LOTUS\DOMINO\xml-apis.jar
QPJC5=C:\LOTUS\DOMINO\log4j-118compat.jar
QPJC6=C:\LOTUS\DOMINO\ibmjsse.jar
QPJC7=C:\LOTUS\DOMINO\STCore.jar
QPJC8=C:\LOTUS\DOMINO\STMtgManagement.jar
QPJC9=C:\LOTUS\DOMINO\cs.jar
```

Note: This example also includes the cs.jar file added in the steps in "Configuring qpservlet" on page 101. It is not necessary to have this for the meeting functionality to work.

Specifying the Web Conferencing authentication name

To specify the Web Conferencing authentication name, complete the following steps:

1. Open the qpconfig.xml file created in 4.4.7, "Configuring the My Team Workplace portlet" on page 101 in a text editor.

2. Scroll down to the Sametime section.

   Remove the following lines from the beginning and end of the <Search_Places> section, respectively:

   ```
   <!-- =============== START OF SAMPLE =================
   =============== END OF SAMPLE =================== -->
   ```

3. Modify the Search Places tags for your environment. Example 4-9 on page 114 shows our example.

Example 4-9   The qpconfig.xml file for the Online Meetings section

```
<sametime local_users="false" ldap="true">
      <meetings invite_servers="false">
         <tools>
            <audio enabled="true"/>
            <video enabled="true"/>
         </tools>
         <credentials>
            <dn>cn=domino admin/o=itso</dn>
            <password>passw0rd</password>
         </credentials>
      </meetings>
   </sametime>
```

Note: The user you specify in credentials <dn> and <password> must satisfy the following conditions:

► The user should exist only in the Domino Directory of Sametime; the user should not be listed in the LDAP used by WebSphere Portal or Sametime.

► The user should be an administrator of Sametime.

To test this, go to:

```
http://sametime.domain.com/stcenter.nsf
```

Click **Administer the Server**. For the user name and password that you enter here, you will need to enter the Domino canonical user name and password into the credentials section of the qpconfig.xml file.

4. Click **Save and Close** to save the XML file.

Specifying Sametime Community server in Team Workplace

To specify the Sametime Community server in Team Workplace, complete the following steps:

1. Open a browser and enter the URL of the Team Workplace server administration console (http://kingston.itsc.austin.ibm.com/quickplace in our environment).

2. Click **Sign In** on the left side of the page.

3. Enter the user name and password of a Team Workplace server administrator (qpadmin in our environment).

4. Click **Server Settings** in the table of contents.

5. Click **Other Options** in the table of contents.

6. Click **Edit Options**.

7. Under Sametime Servers, type the full URL of the Sametime meeting server (http://laredo.itsc.austin.ibm.com in our environment).

8. Click **Next**.

9. Restart the Team Workplace server for the changes to take effect.

To test a user's ability to create an online meeting, complete the following steps:

1. Sign in to a place you have created on the Team Workplace server.

2. Click **New**.

3. Select **Online Meeting**. Click **New**.

4. Give the meeting a name, and click **Publish**.

5. Next, leave the default to leave the page **On Calendar Only**. Click **Next**.

6. This will take you to the calendar view. Click the meeting you just created. You should see something similar to Example 4-10, which is the test meeting we created in our environment.

Example 4-10   Error creating meeting in Team Workplace

```
This Meeting has not started.

(This meeting is not password protected.)

This meeting is located at the following address (URL).
http://laredo.itsc.austin.ibm.com:8088/stconf.nsf/meeting/18218230E595712E86256
EEF0069F8EC?OpenDocument
```

If you see an error stating that the meeting was not created, see the Technote *Knowledge Collection: QuickPlace Issues Related to Sametime*, 1115409, to help you troubleshoot the problem, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21115409

4.5 Placing portlets on a page for testing

To use the Domino portlets, you would deploy them on a page and enable users to access the portlet just as you would any other portlet. The only difference is the Notes and Domino portlets. These portlets consist of the Lotus Notes View, Lotus Notes Mail, My Lotus Notes Mail, My Lotus Notes Calendar, My Lotus Notes To Do, Lotus Notes Teamroom, and Lotus Notes Discussion. These

portlets install as one portlet, by default the Lotus Notes View portlet type. If you want to use any of the other portlet types, complete the following steps:

1. Log in to WebSphere Portal as the Portal administrator.
2. Click **Administration**.
3. Select **Portlets → Manage Portlets**.
4. Select the **Lotus Notes View** from the Portlets picker.
5. Click the **Copy** button.
6. After the portlet has been copied, select the copy and click **Modify Parameters**.
7. In the Portlet Type field, use the values in Table 4-4 to determine what portlet type you need.

Table 4-4   Notes and Domino portlet types

Portlet	Portlet type
Lotus Notes View	NOTESVIEW
Lotus Notes Mail	NOTESMAIL
My Lotus Notes Mail	MYINBOX
My Lotus Notes Calendar	MYCALENDAR
My Lotus Notes To Do	MYTODO
Lotus Notes Discussion	NOTESDISCUSSION
Lotus Notes Teamroom	NOTESTEAMROOM

8. Click **Save**.
9. Select the radio button next to the language to which you would like to change the title.
10. Select **Set title for selected locale** and enter the title you would like for your portlet type.
11. Click **Save**.
12. Click **Cancel** to return to the Manage Portlets page.

4.6 Known problems and fixes in this configuration

The following list describes known problems with hotfixes available from Lotus Technical Support:

► Sametime contact list

Lotus Collab Portlets: Sametime Contact List Portlet Sporadically Shows "Contact List Is Empty", SPR CPRE5Z7SDG, Technote 1174300, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174300

► Awareness in WebSphere Portal or Team Workplace

Sporadic Error on Portal Pages with "Who Is Here" Portlet: "Failed to Contact Messaging Server", SPR CPRE5ZSGHP, Technote 1174296, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174296

Sametime Awareness in Lotus Collaborative Portlets Does not Function Consistently, SPR CPRE5ZSGNJ, Technote 1174303, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174303

► My Team Workplace

– *"My Tasks" Link in Lotus Team Workplaces Portlet Displays "0" Instead of Correct Title*, SPR JLIN5Y783D, Technote 1174645, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174645

– *"What's New" Link in 'My Lotus Team Workplaces' Portlet Shows Author's Full DN rather than Common Name*, SPR BGAR5ZM754, Technote 1174643, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174643

► Who Is Here

QuickPlace Awareness Causes 'Who Is Here' Portlet To Show Names Multiple Times, SPR CPRE5ZBRSY, Technote, 1174649, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174649

► Lotus Web Conferencing

– *Sametime Meetings Created from Lotus Web Conf Portlet or QuickPlace Connect to Wrong Port*, SPR CPRE5Z4KRM, Technote 1174639, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174639

– *Meeting Center Link in Lotus Web Conferencing Portlet Does not Include Port Number*, SPR BJAS5WJ7DH, Technote 1177876, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177876

► Awareness in WebSphere Portal or Team Workplace

Sun JVM Causes Browser to Hang on QuickPlace and Portal Server Pages, SPR CPRE5Z8HYK, Technote 1174295, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174295

► Team Workplace

– *SSO to QuickPlace Does not Work in Dual Directory Env; Causes Problems with Portlets*, SPR SSHD5WGKAY and SPR SSHD62XRHM, Technote 1177890. QuickPlace Hotfix 34 fixes the problem to any place of which you are a member. If you are a member of a group that has access to the place, or can access the place because the place allows anonymous access, you will continue to have SSO problems. See:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177890

– *Dual Directory Env: My Team Workplaces Portlet Does not Return List of Places You Are a Member of*, SPR CPRE646P6X, Technote 1177882, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177882

– *Dual Directory Env: Accessing a Team Workplace You Are a Member of, After Signing into Portal, Breaks Awareness*, SPR CPRE645S6B, Technote 1177881, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177881

► Awareness in WebSphere Portal

In Dual Directory Environment Awareness Does not Work in People Finder Portlet, SPR CPRE647HUT, Technote 1177879, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177879

► Instant Messaging and Web Conferencing

Dual Directory Env: Sametime Does not Work Behind TAM Junction; Web Conferencing Portlet Does not Function Properly, SPR CPRE645RMC, Technote 1177874, available at:

http://www.ibm.com/support/docview.wss?rs=0&uid=swg21177874

5

Setting up secure communication

This chapter describes how to secure the communication of the collaborative portal solution. We organize the solutions into the following topics:

► SSL implementation scope

► Enabling SSL on Domino-based products

► Enabling SSL on the IBM Directory Server

► Enabling SSL on the WebSphere Portal server

► SSL communication with IBM Directory Server

► SSL between the WebSphere Portal and Domino applications

► SSL between Team Workplace and Instant Messaging and Web Conferencing

5.1 SSL implementation scope

In this chapter, we discuss the implementation of the SSL protocol for communication between the components of the collaborative WebSphere Portal server. Figure 5-1 shows this implementation scope. The SSL implementation will be for Hypertext Transfer Protocol (HTTP), Distributed IP Inter Object Protocol (DIIOP), and Lightweight Directory Access Protocol (LDAP).
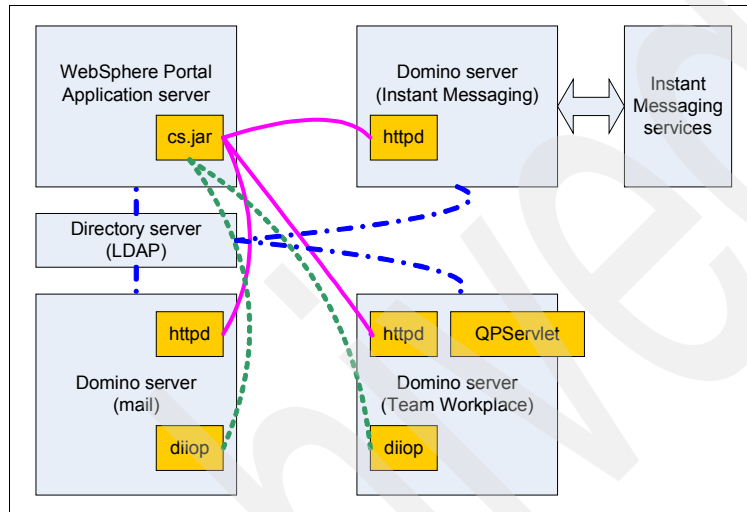
Figure 5-1   Component interaction and interconnection

In Figure 5-1, the connections are shown as lines between the components:

► The HTTP connection is shown with a straight line ——.

► The LDAP connection is shown with a dash-dot line — · —.

► The DIIOP connection is shown with a dotted line · - - ·.

The SSL configuration and implementation requires that:

► All components should have certificates.

► All components should trust the certificate authority that generates the certificates.

► Interconnected components should exchange trust using the certificate.

In this example, we use the Domino built-in facility for acting as a certificate authority. In a production implementation, you would use a commercial certificate authority to generate the certificates.

In the first three sections of this chapter, we discuss the implementation of the certificate authority, certificate creation, and SSL enablement for each components. These topics are:

► Enabling SSL on Domino-based products

► Enabling SSL on the IBM Directory Server

► Enabling SSL on the WebSphere Portal server

In the subsequent sections, we discuss the configuration of the interconnections for trust and to let the components know that their partner is now using SSL:

► SSL communication with IBM Directory Server:
   – Enabling SSL for Lotus Team Workplace
   – Enabling SSL for Lotus Instant Messaging and Web Conferencing

► SSL between the WebSphere Portal and Domino applications

► SSL between Team Workplace and Instant Messaging and Web Conferencing

5.2 Enabling SSL on Domino-based products

SSL in Domino-based products can be implemented using self-signed certificates or using a certificate authority (CA) such as VeriSign. Domino can also be configured to be a certificate authority. In our example, we configured Domino to be a certificate authority to mimic the configuration used in a production environment using a CA company such as VeriSign and also to give us the ability to encrypt the LDAP traffic across the environment. Another important reason that we did this was because the Domino Extended Products require that all certificates have to be signed by the same CA for SSL encryption to work. We need to create a Domino CA and sign all certificates with that CA.

We complete the following steps in order to enable SSL across the environment:

► Configuring the Domino certificate authority

► Enabling SSL on additional Domino servers

► Enabling SSL on Lotus Team Workplace

► Enabling SSL on Lotus Instant Messaging and Web Conferencing

5.2.1 Configuring the Domino certificate authority

A Domino certificate authority (CA) server hosts the Domino Certificate Authority application. Most organizations need only a single Domino CA server. To set up a Domino CA server, you must perform these tasks:

1. Load the HTTP task using the `load http` command from the Domino server. See "Activating server processes for Domino" on page 85 for more details.

2. Create the Domino 5 Certificate Authority application:

   a. Using a Notes client, select **File → Database → New**. We create a new database called ca.nsf with the advanced template Domino Certificate Authority (6) server in our kingston server. Click **OK** when the New Database window opens, as shown in Figure 5-2.
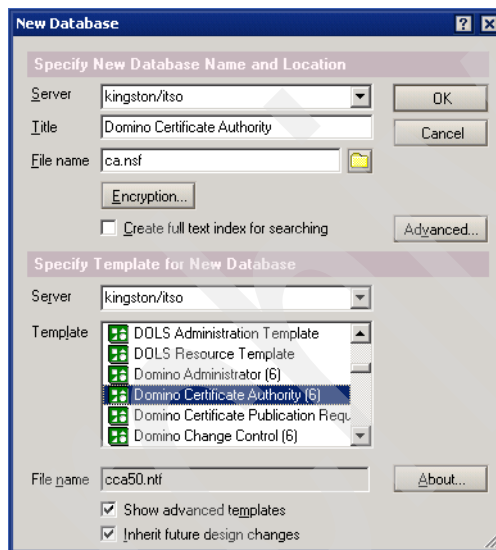
Figure 5-2   New Certificate Authority database

   b. In the Certificate Authority database, select **File → Database → ACL**. Edit the ACL of the Domino 5 Certificate Authority database, as shown in Figure 5-3 on page 123:

      • Add the names of the administrators who will issue and manage Internet certificates (domino admin in our example). Assign the Editor with **Delete** access or **Manager** access and the [CAPrivlegedUser] role to each administrator.

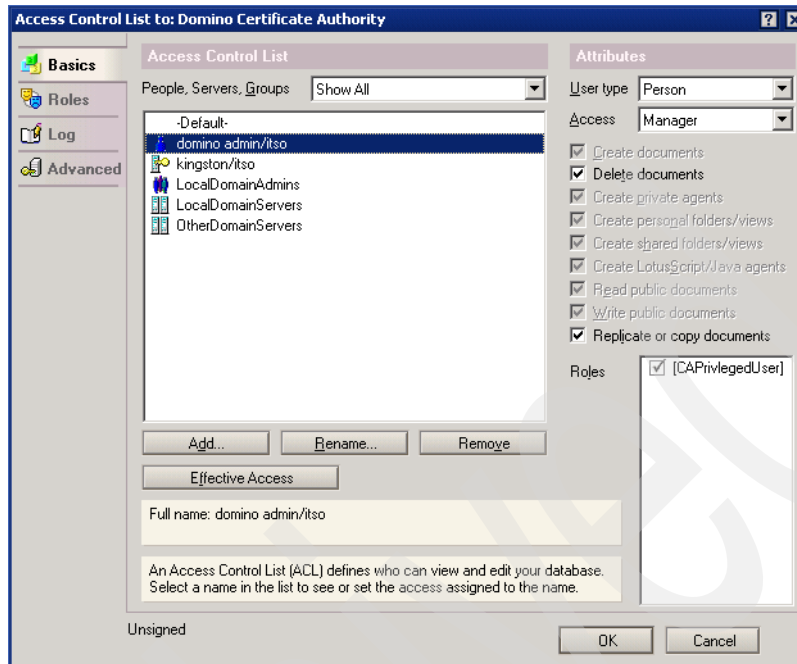   • Set the **Default** access to the Author with Create documents privilege.

Figure 5-3   Administrator ACL to ca.nsf

c. After making changes to the ACL, you will need to close and reopen the database for the change to take effect. Figure 5-4 on page 124 shows the initial page for the Certificate Authority database.
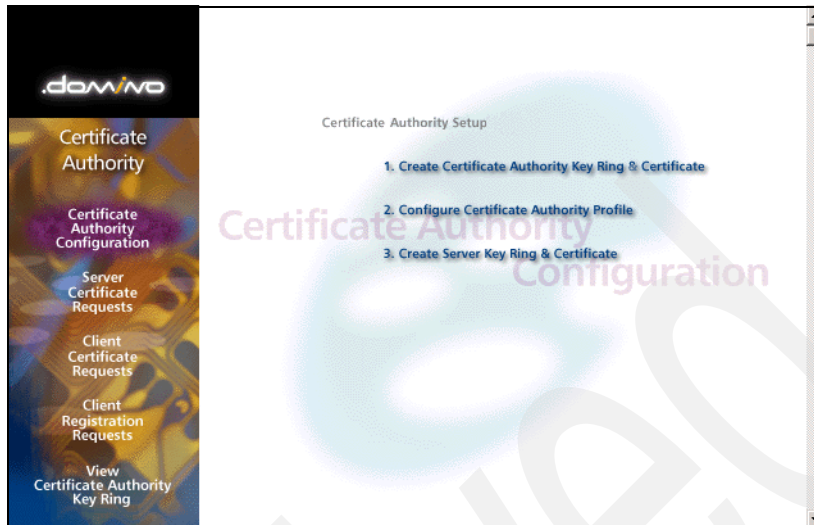
Figure 5-4   Domino Certificate Authority application

3. Create a CA key ring file and CA certificate.

   When you use the Domino administrator to create the CA key ring file, it is
   stored by default in the client's data directory. Make sure that you keep the
   key ring file in a secure location, especially if you copy it to a shared location.
   Only the administrators that you specify should have access to the CA key
   ring file and password.

   a. Click **Create Certificate Authority Key Ring & Certificate**.

   b. Complete the fields in a similar manner as our example, as shown in
      Figure 5-5 on page 125.

Figure 5-5   Create CA key ring file

c. Click **Create Certificate Authority Key Ring**. After you review the information about the key ring file and CA name, click **OK**.

d. Make a backup copy of the certificate authority key ring file, and store it in a secure location.

4. Configure the CA profile to specify the key ring and mail settings.

   The Domino Certificate Authority application profile identifies the CA's key ring file and specifies the name of the CA server. Domino adds a link to the CA server when you send a message to clients and server administrators who request certificates. The clients and server administrators use this information to determine where to pick up certificates.

   a. Click **Configure Certificate Authority Profile**.

   b. If necessary, enter the CA key ring path and file name in the **CA Key File** field. By default, Notes looks for the key ring file on the local hard drive. You can also specify a network drive accessible to other administrators.

   c. Enter the TCP/IP DNS name of the server that runs the CA application in the **Certificate Server DNS Name** field (kingston.isto.austin.ibm.com in our example). Domino uses this name to indicate where to pick up signed certificates in the messages sent to administrators and clients.

d. Configure the remaining fields as you see fit for your environment. Figure 5-6 shows our example.

Figure 5-6   Certificate Authority Profile example

e. Click **Save & Close**.

5. Set up SSL on the CA server.

Because server administrators and clients use browsers to access the CA server to request and pick up certificates, use SSL to protect the CA server. When you set up the CA server for SSL, you create the server key ring file

and request a server certificate. Domino automatically approves the server certificate and merges the CA certificate as a trusted root.

a. Click **Create Server Key Ring & Certificate**.

b. Complete the fields in a similar manner as our example, as shown in Figure 5-7.

Figure 5-7   Create CA Server key ring example

c. Click **Create Server Key Ring**.

d. Enter the CA key ring file password, and then click **OK**. The server SSL key ring file is created.

e. Copy the server key ring file and put the file in the Domino data directory on the server. The Domino Certificate Authority application creates the file locally; however, the server needs the key ring file to use SSL.

f. Close the Domino Certificate Authority application.

> **Note:** If you did not name the key file keyfile.kyr, you can change the name the Domino server looks for by opening the Server document in the Name and Address book. Click the **Ports** → **Internet Ports** tab, and update the SSL key file name field.

6. Configure the HTTP task for SSL on the Domino CA server:

a. From the Domino Administrator, click **Configuration** → **Servers**, and open the Server document for the Domino CA server.

b. Click the **Ports** → **Internet Ports** → **Web** tab.

c. Disable **TCP/IP port status** and enable **SSL port status**.

d. Make sure to set Name & Password field to **Yes**.

e. Click **Save and Close**.

f. Restart the Domino server.

Now, the Domino Certification Authority server is configured and it will listen for HTTP requests over port 443 only.

5.2.2 Enabling SSL on additional Domino servers

SSL on a Domino server enables that connected clients and servers use SSL to ensure privacy and authentication on the network. You set up SSL on a protocol-by-protocol basis. For the collaborative portlets, you can secure any one of the following protocols: HTTP, LDAP, and DIIOP.

In the following sections, we explain how to set up certificates on the Domino server, enable the server task to use SSL, configure the Domino server to be able to secure a port, and secure each of the ports used with the collaborative portlets. We also include what features of the portlets will now be secured.

Setting up certificates on the Domino server

To set up SSL on your server, you need a key ring containing a server certificate from a certificate authority. You can obtain a server certificate from a certificate authority (CA) and then install it in a key ring. A server certificate is a binary file

that uniquely identifies the server. The server certificate is stored on the server's hard drive and contains a public key, a name, an expiration date, and a digital signature. The key ring also contains root certificates used by the server to make trust decisions.

Here, we describe the process to follow if you need to set up SSL on a Domino server that is not already a Domino certificate authority server. We explain the steps needed to be performed by a CA administrator on the Domino CA server created in 5.2.1, "Configuring the Domino certificate authority" on page 122.

To set up certificates on a Domino server, complete the following steps:

1. Set up the Server Certificate Admin application (CERTSRV.NSF), which Domino creates automatically during server setup.

   Domino automatically creates the Server Certificate Admin application during server setup. If the Server Certificate Admin application is not available after you start the Domino server, use the Server Certificate Admin template (CSRV50.NTF) to create it. Open the certsrv.nsf by selecting **File →**
   Database → Open from Notes client. Check its ACL by selecting **File →**
   Database → Access Control and ensure that the Default Access is set to No Access and the Domino admin's name shows with Manager access.

   You will need to close and reopen the database if you make any changes to the ACL. Figure 5-8 shows the navigator for the Server Certificate Administration database.
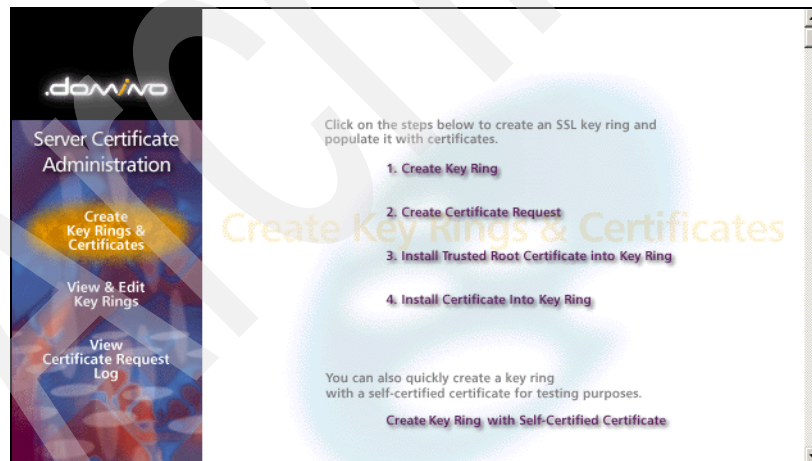
Figure 5-8   Certificate Server navigator menu

2. Create a server key ring file to store the server certificate.

Before you request a certificate from a CA, you must create a key ring file to store the certificates. A key ring file is a binary file that is password protected and stored on the server's hard drive. When you create a server key ring file (.kyr), Domino generates an unsigned server certificate and automatically includes several trusted root certificates. The unsigned server certificate is not valid until it is signed by a certifier. Domino also creates a stash file (.sth) using the same name as the key ring file. Domino uses the stash file to store the key ring file password for unattended access to the server key ring file.

Every server certificate includes a distinguished name used for SSL connections. You set up this distinguished name when you create the server key ring file. Some components of a distinguished name are optional; however, the more components you include, the less likely you are to encounter an identical name elsewhere on the Internet.

To create a server key ring file:

a. Click **Create Key Ring** from the navigator, as shown in Figure 5-8 on page 129.

b. Complete the fields for your environment. Figure 5-9 on page 131 shows our environment.

Figure 5-9   Create Key Ring file

    c. Click **Create Key Ring**.

    d. After you read the information about the key ring file and distinguished name, click **OK**. Notes creates the key ring file and stash (.sth) file and places them in the Notes data directory on the client machine used to create the key ring.

3. Request an SSL server certificate from the CA.

When you request an SSL server certificate, use Public-Key Cryptography Standards (PKCS) format, an industry-standard format that many CAs, including Domino, understand. Before you request a certificate from a third-party CA, make sure the CA uses the PKCS format. A certificate request is essentially certificate data that has not been signed by a CA. The CA turns the request into a certificate by signing it.

If you are requesting a server certificate from a server-based certification authority, you can use the Notes client to create the server key ring and the server certificate in the Certificate Requests database. You must be able to access the Domino server using the Notes client.

To request a server certificate from a Domino CA using a Web browser:

a. Click **Create Certificate Request** and complete the fields in a similar manner to the fields shown in Figure 5-10.

Figure 5-10   Create Server Certificate Request

b. Click **Create Certificate Request** and enter the password for the server key ring file.

c. Copy the certificate request to the system Clipboard (include the Begin Certificate and End Certificate lines), and click **OK**.

d. Open a Web browser to the Domino CA server's Certificate Authority database (`https://kingston.istc.austin.ibm.com/ca.nsf` in our environment).

e. Click **Request Server Certificate** and enter your name, e-mail address, phone number, and any comments for the CA.

f. Paste the certificate request into the dialog box, and then click **Submit Certificate Request**.

4. Merge the CA certificate as a trusted root into the server key ring file:

a. From the Web browser, click **Accept This Authority in Your Server** and highlight the certificate text and copy it to the system Clipboard (include the Begin Certificate and End Certificate lines).

b. From the Notes client, open the Server Certificate Admin application on the server you are enabling for SSL.

c. Click **Install Trusted Root Certificate into Key Ring** from the navigator.

d. Fill in the name of the key ring file, certificate label, and paste the Clipboard into the Certificate Source field. Click **Merge Trusted Root Certificate into Key Ring**.

e. Enter the password for the key ring file, and then click **OK**. Our example is shown in Figure 5-11 on page 134.

Figure 5-11   Install Trusted Root Certificate

5. The CA approves the request for a server certificate and sends notification that you can pick up the certificate:

   a. The Domino CA administrator should open the Domino Certificate Authority database using Notes client.

   b. Click **Server Certificate Requests** from the navigator shown in Figure 5-4 on page 124.

   c. Double-click the request just submitted, assign a validity period, and copy the pickup ID to the Clipboard. Click **Approve**.

6. Pick up the signed server certificate. Now that the Domino CA administrator has approved this server, you need to pick the certificate up, and merge it into your key ring file:

   a. Open Microsoft Internet Explorer and browse to the Domino CA server's Certificate Authority database (in our environment, `https://kingston.istc.austin.ibm.com/ca.nsf`).

   b. Click **Pick up Server Certificate** and enter the pickup ID you just copied and select **Pick Up Signed Certificate**.

   c. Highlight the certificate text and copy it to the system Clipboard (include the Begin Certificate and End Certificate lines).

7. Merge the approved server certificate into the key ring file:

   a. From the Notes client, open the Server Certificate Admin application.

   b. Click **Install Certificate into Key Ring**.

   c. Enter the file name for the key ring that will store this certificate. You specified this key ring file when you created the server certificate request.

   d. In the Certificate Source field, choose **Clipboard**. Paste the Clipboard contents into the next field, as shown in our example in Figure 5-12 on page 136.
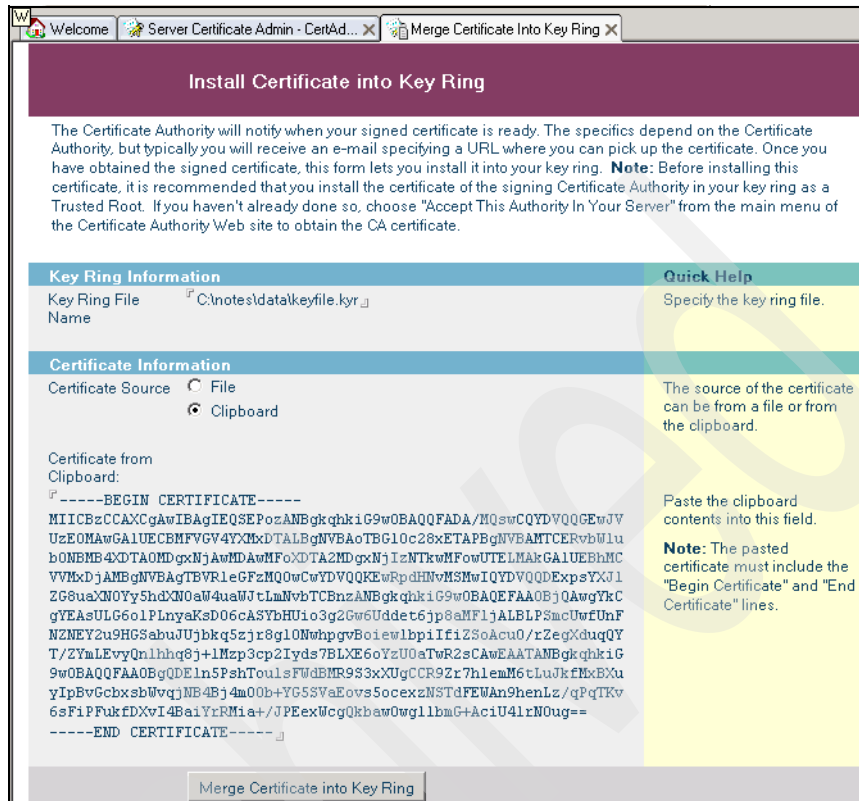
Figure 5-12   Install Certificate into Key Ring

   e. Click **Merge Certificate into Key Ring**.

   f. Enter the password for the key ring file, and then click **OK** to approve the merge.

Finally, copy the server key ring file and put the file in the Domino data directory on the server (C:\Lotus\Domino\Data on kingston/itso in our example).

Note: If you did not name the key file keyfile.kyr, you can change the name the Domino server looks for by opening the Server document in the Name and Address book. Click the **Ports** → **Internet Ports** tab, and update the SSL key file name field.

Enabling server tasks to use SSL

The tasks that we enable for SSL are HTTP, DIIOP, and LDAP tasks in Domino servers. To enable the server tasks to use SSL, complete the following steps:

1. From the Domino administrator, click **Configuration** → **Servers**, and open the Server document for the Domino server.

2. Click **Ports** → **Internet Ports**; depending on the server that you configure, you can select the following tabs:

HTTP	Web tab
DIIOP	DIIOP tab
LDAP	LDAP tab

3. Disable **TCP/IP port status** and enable **SSL port status**. This will enable the following ports:

HTTP	443
DIIOP	63149
LDAP	636

4. Make sure to set Name & Password field to **Yes**.

5. Click **Save and Close**.

6. Restart the Domino server.

5.2.3 Enabling SSL on Lotus Team Workplace

To enable SSL on the Lotus Team Workplace server, you need to enable SSL on the base Domino server. After enabling SSL on the Team Workplace server, the links from My Places and from the Team Workplace portlet will continue to open the places at non-SSL URLs. To correct this, you will need to perform the following steps:

1. Update the place catalog on the Team Workplace server:

   a. Use a Notes client, and open the placecatalog.nsf database from the Team Workplace server by selecting **File** → **Database** → **Open**.

   b. Select the **PlaceServers** view and edit the Server document.

   c. Change the PlaceServerAccessProtocol to HTTPS.

   d. Change the PlaceServerAccessTCPPort to 443 (if that is the SSL port you are using).

   e. Save and close the document.

2. Correct Search all places within the My Team Workplaces portlet by editing the qpconfig.xml file so that it will connect to the server over SSL:

   a. Open the qpconifg.xml file in a text editor. In our example, we have this file in the Domino data directory.

b. Scroll down to the search_places section and update it similar to that shown in Example 5-1.

Example 5-1   Setting qpconfig.xml to search places for SSL

```
<search_places enabled="true" log_level="0" anonymous="true">
    <domain_catalog_server ssl="true">
          <port>443</port>
          <domino_server_name>kingston/itso</domino_server_name>
          <path_prefix></path_prefix>
          <hostname>kingston.itso.austin.ibm.com</hostname>
    </domain_catalog_server>
</search_places>
```

c. Save and close the qpconfig.xml file.

d. Restart the Team Workplace server HTTP task for the change to take effect.

5.2.4 Enabling SSL on Lotus Instant Messaging and Web Conferencing

To enable SSL on the Lotus Instant Messaging and Web Conferencing server, you first need to enable SSL on the Domino server. Additional steps need to be performed to set up SSL access for the Sametime servlets.

After you force all HTTP connections to the Instant Messaging and Web Conferencing server to use SSL, you must also configure the Sametime server to support SSL for HTTP connections to its servlets. If the Domino HTTP server is configured to require SSL for all connections, and the Sametime server is not configured to support SSL connections to its servlets, the Sametime server will not function properly and users will be unable to access the server.

To ensure that Sametime servlets can be accessed using SSL when Domino requires SSL for all connections, you must perform the following steps:

1. Obtain the appropriate SSL trusted root or SSL server certificate.

   When the Domino server is set up to use SSL, an SSL server certificate is received from a certification authority (CA) and merged into the Domino Server Certificate Admin (certsrv.nsf) database.

   You need a copy of the CA certificate in the local disk used with IBM Key Management tools. This certificate needs to be in the format of Base64 encoding. One of the options is to import this in the Microsoft Internet Explorer trusted certificate authority and export the certificate.

Perform the following to import a CA certificate:

a. Open the Instant Messaging and Web Conferencing server Web page using Microsoft Internet Explorer. (Our URL is `https://laredo.itsc.austin.ibm.com`.)

b. The first time you access this site, you will see a Security Alert dialog box. Select **View Certificate** and then in the Certificate dialog box, select the **Certification Path** tab. Click the certificate authority above the server, as shown in Figure 5-13.

Figure 5-13   Select CA on Certification Path tab

c. Click **View Certificate**, and then in the Certificate dialog box, click **Install Certificate**. Select the **Automatically select the certificate store based on the type of certificate** option.

d. You should see a Security Warning similar to the one shown in Figure 5-14 on page 140. Accept this certificate by clicking **Yes**.

Figure 5-14   Security Warning dialog box

   e. You should see a message indicating that the SSL server certificate was imported successfully. Click **OK** to close this dialog box.

Perform the following to extract the CA certificate:

   a. From Microsoft Internet Explorer, select **Tools** → **Internet Options**.

   b. Click the **Content** tab. Click the **Certificates** button and select the **Trusted Root Certification Authorities** tab.

   c. Scroll down the list of certificates and select the certificate authority certificate that you imported earlier in this procedure, as shown in our example in Figure 5-15 on page 141.
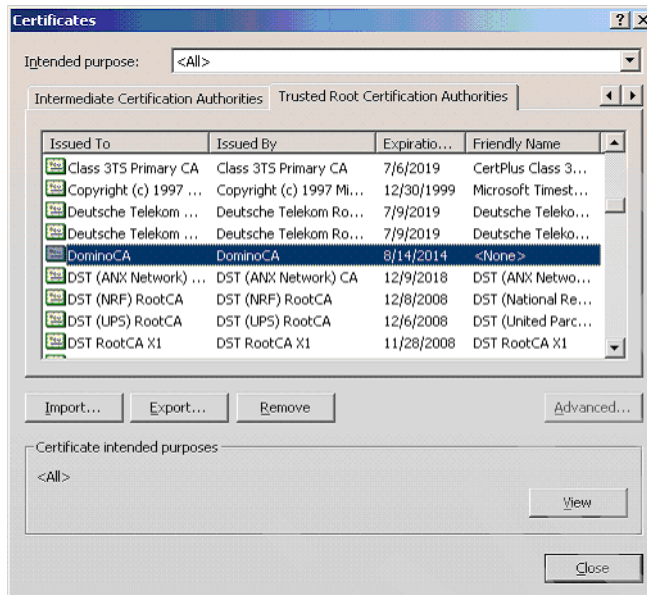
Figure 5-15 Getting the CA certificate

    d. Click the **Export** button. Export the certificate in the format of Base64 encoded X.509 (.cer) and put it in a file such as cacert.cer.

    e. Close all remaining dialog boxes.

2. Install IBM Key Management (KeyMan). The KeyMan utility must be installed before you can create the key store token needed to store the SSL certificate for SSL connections to Sametime servlets. The KeyMan utility is installed using the executable keyman-1_43.exe file located in the Instant Messaging and Web Conferencing program directory C:\Lotus\Domino. Follow the installation wizard and accept the default values.

3. Use the IBM KeyMan program to create a key store token on the Sametime server. The KeyMan key store token stores the SSL certificates required to ensure that the Sametime servlets can be accessed using SSL. In this procedure, you create a KeyMan key store token named "stkeys.pfx" and store this token in the Instant Messaging and Web Conferencing directory C:\Lotus\Domino.

    a. Start the IBM KeyMan by selecting **Programs → IBM KeyMan → KeyMan**. Figure 5-16 on page 142 shows the initial window.

Figure 5-16   Start the IBM KeyMan program

   b. At the KeyMan: New/Open window, click the **Create new** icon (located on
      the left side of the window).

   c. At the KM: New window, select the **PKCS#12 Token** (password
      protected) option. Click the green check mark to continue.

   d. A newly-created token appears. Select **File → Save** to save the token.

   e. At the KM: Save token window, enter and then re-enter the passphrase
      that you will use to protect this key store token. You will be required to
      enter this password any time you open this token to manage SSL
      certificates. Click the purple arrow to continue.

   f. At the KM Save token...Save PKCS#12 Token window, complete the
      following fields:

      • Save to file: This needs to be called `stkeys.pfx`.
      • File format: Accept the default value of **PKCS#12 / PFX**.

   g. Select the green check mark to continue.

4. Import the appropriate SSL trusted root certificate or SSL server certificate to
   the key store token. We use the cacerts.cer file that we created from the Web
   browser.

   a. Open the key store token stkeys.pfx file. You need to supply the
      passphrase.

   b. When the stkeys.pfx IBM KeyMan token opens, select **File → Import** and
      import the cacert.cer file that we exported previously.

   c. At this point, the SSL certificate is imported into the KeyMan key store
      token. To verify that the certificate was imported successfully, select
      Trusted CA Certificates from the drop-down list in the KeyMan key store
      token, as shown in Figure 5-17 on page 143.
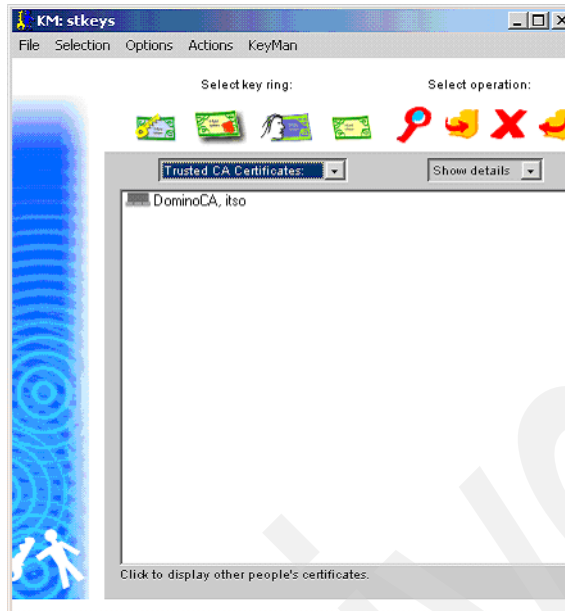
Figure 5-17   Trusted CA Certificates

   d. Save the token by selecting **File** → **Save**, and close KeyMan.

5. Configure the sametime.ini file on the Sametime server.

    The sametime.ini file on the Sametime server contains parameters that must
    be configured appropriately to ensure that HTTP connections to the
    Sametime servlets can be encrypted with SSL.

   a. Use a text editor to open the sametime.ini file in C:\Lotus\Domino.

   b. In the [Config] section of the sametime.ini file, alter or add the settings
       similar to those shown in Example 5-2.

> **Note:** These setting are case sensitive, so make sure you enter them
> with the correct case.

Example 5-2   Changes in the sametime.ini file for SSL

```
ConfigurationPort=443
ConfigurationSSLEnabled=true
SSLManagerClassName=com.lotus.sametime.configuration.IBMJSSE118Manager
javax.net.ssl.keyStore=c:\lotus\domino\stkeys.pfx
javax.net.ssl.trustStore=c:\lotus\domino\stkeys.pfx
javax.net.ssl.keyStorePassword=passw0rd
javax.net.ssl.trustStorePassword=passw0rd
ConfigurationHost=laredo.itsc.austin.ibm.com
```

c. Save and close the sametime.ini file.

d. Restart the Instant Messaging and Web Conferencing Domino server.

5.3 Enabling SSL on the IBM Directory Server

On the IBM Directory Server, you have three choices for using Secure Sockets Layer (SSL):

SSL Off   Clients are permitted to conduct only unsecure communications.

SSL On   Clients are permitted to conduct either secure or unsecure communications.

SSL Only   Clients are not permitted to conduct unsecured communications. This is the most secure way to configure your server.

If you choose to use either type of SSL communications, you have two choices for SSL authentication on the LDAP server:

Server authentication   For server authentication, the IBM Directory Server supplies the client with the server's X.509 certificate during the initial SSL handshake. If the client validates the server's certificate, a secure, encrypted communication channel is established between the server and the client application.

Server and client authentication

This type of authentication provides for two-way authentication between the LDAP client and the LDAP server. With client authentication, the LDAP client must have a digital certificate (based on the X.509 standard). This digital certificate is used to authenticate the LDAP client to the server.

For server authentication to work, the IBM Directory Server must have a private key and associated server certificate in the server's key database file. We configured the SSL On option with SSL authentication set to Server authentication. We followed these steps:

1. Configure the GSKit iKeyman utility. This utility is installed with IBM Directory Server. You need to perform the following steps:

   a. Back up C:\IBM\Java131\jre\lib\security\java.security.

   b. Copy C:\IBM\gsk7\classes\gsk_java.security to C:\IBM\Java131\jre\lib\security\java.security.

c. Copy all JAR files from C:\IBM\gsk7\classes\jre\lib\ext to C:\IBM\Java131\jre\lib\ext.

d. Set JAVA_HOME to C:\IBM\Java131\jre in your system environment.

2. Start iKeyman by running the `gsk7ikm` command from C:\IBM\gsk7\bin.

3. Create a new CMS key database file by selecting **Key Database File** → **New** and give it a name and password. We saved it as C:\IBM\LDAP\etc\idskey.kdb. Make sure that Set expiration time is not selected and that Stash the password to a file is selected.

4. Click **Create** → **New Certificate Request**, and fill in the fields for your environment. Save it. We entered:

    – Key Label: `IDSkey`
    – Common Name: `phoenix.itsc.austin.ibm.com`
    – Organization: `itso`
    – Country: **US**

5. Get the certificates signed by CA and import them into your key file. The certificate needs to be submitted and signed by a CA. We use the Domino CA, and this procedure is similar to the certificate request for the Domino server discussed in 5.2.2, "Enabling SSL on additional Domino servers" on page 128.

    a. Use the certificate request that is created in the previous step for the Request Server Certificate entry.

    b. Save the CA's certificate text as an ARM file, such as dominoca.arm. Using the iKeyman utility, select **Signer Certificates** from the drop-down list and import the dominoca.arm file.

    c. After the certificate request is approved, pick up the server certificate and put it in another ARM file, such as idssigned.arm. Using the iKeyman utility, select **Personal Certificates** from the drop-down list and receive the idssigned.arm file.

6. Close the iKeyman utility.

7. Enable SSL on the IBM Directory Server. Now that you have created the signed key file, you need to enable SSL in the IBM Directory Server:

    a. Open a browser to the LDAP Web Administration page (in our example, `http://phoenix.itsc.austin.ibm.com:9080/IDSWebApp/IDSjsp/Login.jsp`).

    b. Sign in as the LDAP administrator (cn=root in our example).

    c. Select **Server Administration** → **Manage Security Properties**:

        • Under Enable secure connections choose **SSL**.
        • Under Authentication method select **Server authentication**.

d. Click **Apply**.

e. Click the **Key Database** tab. Enter the Key database path and file name, password, and Key Label. Click **Apply**.

f. Click **OK** to save your changes and close the LDAP Web Administration page.

g. Restart the IBM Directory Server for you changes to take effect.

5.4 Enabling SSL on the WebSphere Portal server

This section describes the SSL setup for the WebSphere Portal server machine. The setup is divided into the IBM HTTP Server setup and the WebSphere Application Server setup.

5.4.1 Configuring IBM HTTP Server

To configure SSL for the IBM HTTP Server, complete the following tasks:

1. Enable the httpd.conf file for SSL:

   a. Use httpd.conf.sample as httpd.conf. You might want to back up the original httpd.conf file.

   b. Edit the httpd.conf file under the conf directory and find or update the following lines:

      • ServerName should be the fully qualified host name of the server.

      • Uncomment the following SSL settings, which for our example are:

```
LoadModule ibm_ssl_module/IBMModuleSSL128.dll
Listen 443
<VirtualHost pretoria.itsc.austin.ibm.com:443>
SSLEnable
</VirtualHost>
Keyfile "c:\ibm\IBMHttpServer/ssl/keyfile.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
```

> **Note:** The keyfile path has been modified to include *ssl* instead of keys.

   c. Add WebSphere plug-in directives into the new httpd.conf file:

```
LoadModule ibm_app_server_http_module
"c:\ibm\WebSphere\AppServer/bin/mod_ibm_app_server_http.dll"
WebSpherePluginConfig
"c:\ibm\WebSphere\AppServer/config/cells/plugin-cfg.xml"
```

> **Tip:** The above text is two separate lines. Within the httpd.conf file, the lines are not wrapped.

   d. Save the changes to the httpd.conf file.

2. Create the IBM HTTP Server keystore.

   To create the IBM HTTP Server keystore database used to store certificates, complete the following steps:

   a. Select **Programs** → **IBM HTTP Server 1.3.26** → **Start Key Management Utility**.

   b. From the menu bar, click **Key Database File** → **New**.

   c. Enter the file name and path specification that you entered in the httpd.conf file in step 1b on page 146. The file type should be CMS key database file. Click **OK**.

   d. Enter the password for the key and select the **Stash the password to a file** option. Click **OK**. In the completion window, click **OK**.

   e. Select **Create** → **New Certificate Request**.

   f. Fill in the fields for your environment. In our environment, we entered:

   – Key Label: HTTPkey
   – Common Name: pretoria.itsc.austin.ibm.com
   – Organization: itso
   – Country: **US**

   g. Specify a place to save the file, and remember the name (C:\IBM\IBMHttpServer\ssl\certreq.arm in our example).

   Leave the iKeyman utility open, because you will need it in the following steps.

3. Get the certificates signed by the CA and import them into your key file. Again, we use the Domino CA that we configured. The procedure here is similar to 5.2.2, "Enabling SSL on additional Domino servers" on page 128.

   a. Use the certificate request that is created in the previous step for the Request Server Certificate entry.

   b. Save the CA's certificate text as an ARM file, such as dominoca.arm. Using the iKeyman utility, select **Signer Certificates** from the drop-down list and import the dominoca.arm file.

   c. After the certificate request is approved, pick up the server certificate and put it in another ARM file, such as HTTPsigned.arm. Using the iKeyman utility, select **Personal Certificates** from the drop-down list and receive the HTTPsigned.arm.

4. Close the iKeyman utility.

5. Verify IBM HTTP Server.

   After the IBM HTTP Server SSL configuration, we recommend that you verify the setup. Start the IBM HTTP Server, either from the Windows Services applet or using the command `net start "IBM HTTP Server 1.3.26"`. Verify accessibility to both the HTTP and HTTPS port.

5.4.2 Configuring WebSphere Application Server

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 will then forward the traffic to WebSphere Portal). Next, you update the virtual host list for WebSphere Application Server to include the correct host name and port number and regenerate the plug-in configuration.

> **Note:** For more information regarding 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 steps:

1. Ensure that the WebSphere Application Server server1 is started on the WebSphere Portal server node.

2. Start the WebSphere Application Server Administrative Console from the URL and enter the administrator credential, such as wpsbind (in our case, the URL is `https://pretoria.itsc.austin.ibm.com:9043/admin`).

3. Select **Environment** → **Virtual Hosts** → **default_host** → **Host Aliases**.

4. Create a new host alias for port 443. Click **New** and enter the following values in the New Host Alias page:

   – Host Name: *
   – Port: `443`

5. Click **OK**. Click **Save** and click **Save** again to save to the Master Configuration.

6. Select **Environment** → **Update Web Server Plugin**. Click **OK**.

7. 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 file 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 this to take effect immediately, restart the IBM HTTP Server.

5.4.3 Configuring SSL in WebSphere Portal

To enable SSL in the WebSphere Portal configuration, complete the following steps:

1. From the directory
   C:\WebSphere\AppServer\installedApps\pretoria\wps.ear\wps.war\WEB-INF, in our example, back up the web.xml file to the web.xml.original file.

2. Modify the web.xml file, as shown in Example 5-3. Search `SecurityConstraint_1` to find the appropriate section to modify. Inside, definition change the `<transport-guarantee>` setting from `NONE` to `CONFIDENTIAL`.

> **Note:** The CONFIDENTIAL transport_guarantee allows only HTTPS access to the /myportal/* URLs.

Example 5-3   Our example modified web.xml file 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>
```

3. Save and close the web.xml file.

4. Restart the WebSphere_Portal application server.

5. Restart the IBM HTTP Server.

6. Verify that the WebSphere Portal server is accessible through SSL (for our configuration, we used the URL
   `https://pretoria.itsc.austin.ibm.com/wps/myportal`).

5.5 SSL communication with IBM Directory Server

Because components require LDAP access to extract identity information, they all need to be configured. In the following sections, we discuss the connection configuration to IBM Directory Server over SSL:

- ► Enabling SSL for WebSphere LDAP connections
- ► Enabling SSL for WebSphere Portal LDAP connections
- ► Enabling SSL for Lotus Team Workplace
- ► Enabling SSL for Lotus Instant Messaging and Web Conferencing

5.5.1 Enabling SSL for WebSphere LDAP connections

This section describes how to create the LDAP certificate, import the LDAP server certificate, and configure an SSL repertoire that will be used for LDAP connections by WebSphere Application Server security. To enable SSL for the WebSphere LDAP connection, complete the following steps:

1. Extract the LDAP certificate from the keystore on the IBM Directory Server machine:

   a. Run IBM GSKit V7.0.1.16 from C:\ibm\gsk7\bin directory with the `gsk7ikm` command.

   b. Open the key database file by selecting **Key Database File** → **Open** and select C:\IBM\LDAP\etc\idskey.kdb. See 5.3, "Enabling SSL on the IBM Directory Server" on page 144. Enter the password for the key.

   c. Select **Personal Certificates** from the pull-down menu under Key database content.

   d. Click **Extract Certificate** and put them in a Base64-encoded ASCII data file and click **OK**. We called the extracted file ldapcert.arm.

   e. Close the IBM Key Management window.

2. Import LDAP server certificate into the LdapClientTrustFile.jks file:

   a. Copy the ldapcert.arm file from previous step to the WebSphere Portal server machine.

b. Navigate to the c:\WebSphere\AppServer\bin directory and execute the `ikeyman.bat` command.

c. Create a new JKS key database file called LdapClientTrustFile.jks in C:\WebSphere\AppServer\etc by selecting **Key Database File → New** and supply a password.

d. Select **Signer Certificates** from the pull-down menu under Key database content. Click **Add**.

e. Import the content of the ldapcert.arm file as a Base64-encoded ASCII data and click **OK**.

f. Give this a descriptive label and click **OK**.

g. Close the iKeyman program.

3. Configure SSL using WebSphere Administrative Console (we used the URL `http://pretoria.itsc.austin.ibm.com:9090/admin`).

a. Log in to the Administrative Console as the wpsbind user.

b. Select **Security → SSL**.

c. In the SSL Configuration Repertoires page, click **New**.

d. Enter the following values and click **OK**.

> **Note:** Because we do not use mutual SSL connections for LDAP, we can specify LdapClientTrustFile.jks for both the Key File Name and Trust File Name settings. For mutually authenticated connections, a self-signed certificate would need to be generated for WebSphere Application Server. Key File Name should then point to the key file containing the private key for the generated certificate.

- Alias: `LDAPSSLSettings`
- Key File Name:
  `C:\WebSphere\AppServer\etc\LdapClientTrustFile.jks`
- Key File Password: <password>
- Key File Format: `JKS`
- Trust File Name:
  `C:\WebSphere\AppServer\etc\LdapClientTrustFile.jks`
- Trust File Password: <password>
- Trust File Format: `JKS`
- Client Authentication: Clear
- Security Level: **High**
- Cipher Suites: Do not modify.
- Cryptographic Token: Clear
- Provider: `IBMJSSE`
- Protocol: `SSLv3`

e. Fix the Advanced LDAP Settings.

> **Note:** The following steps are required, because WebSphere Portal configuration scripts incorrectly set the LDAP server type to IBM_Directory_Server while defining the custom search filters that are displayed on the Advanced LDAP Settings page. Clicking **Apply** resets the LDAP server type to Custom, which is the correct setting when custom search filters are used. If these steps are skipped, the search filters will be reset to default values for IBM_Directory_Server. When you click **OK**, you will not be able to access any applications after restarting WebSphere Application Server. To fix the Advanced LDAP Settings:
>
> 1. Select **Security** → **User Registries** → **LDAP**.
> 2. Click **Advanced LDAP Settings**.
> 3. Click **Apply**.

f. Select **Security** → **User Registries** → **LDAP**.

g. Modify the following settings and click **OK**. For our example, we entered the following values:

- Port: 636
- SSL Enabled: Clear
- SSL Configuration: pretoria/LDAPSSLSettings

h. Back on Global Security page, click **Apply** to validate the LDAP connection. Make sure that you do not receive any LDAP validation errors. If there are any errors, correct the problems before saving the configuration; otherwise; WebSphere Application Server will not be able to start.

i. Save the configuration and exit the Administrative Console.

j. Restart the WebSphere Application Server server1 only.

5.5.2 Enabling SSL for WebSphere Portal LDAP connections

To configure WebSphere Portal to connect to the Tivoli Directory Server using SSL, complete the following steps:

1. Import the LDAP certificate to the WebSphere Portal keystore cacerts file.

   WebSphere Portal has no configuration setting to use a specific Java keystore file for secure LDAP connections. Therefore, the certificates needed by WebSphere Portal have to be imported into the JVM default keystore. The JVM default keystore file is called cacerts and it is located in the <WAS_HOME>/java/jre/lib/security directory. By default, the password for this file is changeit.

To import the LDAP server certificate into the WebSphere Portal keystore, complete the following steps:

a. Navigate to the c:\WebSphere\AppServer\bin directory and execute the `ikeyman.bat` command.

b. Select **Key Database File** → **Open** to open the cacerts file from the C:\IBM\WebSphere\AppServer\java\jre\lib\security directory with the type of JKS. Enter the password (the default password is changeit) and click **OK**.

c. Select **Signer Certificates** from the pull-down menu under Key database content. Click **Add** and import the ldapcert.arm file as Base64-encoded ASCII data. Give it an appropriate label and click **OK**.

2. Import the LDAP server certificate into the CSIv2 trust file:

a. Again using the iKeyman application, import the LDAP server certificate. Select **Key Database File** → **Open** to open the JKS file CSIv2ServerTrustFile.jks in the C:\WebSphere\AppServer\etc directory. Enter and confirm the key file password. Click **OK**.

b. Select **Signer Certificates** from the pull-down menu under Key database content. Click **Add** to add the ldapcert.arm file as Base64-encoded ASCII data. Click **OK**. Give it an appropriate label and click **OK**.

c. Close the IBM Key Management window.

3. Configure WMM for LDAP SSL connections.

> **Important:** The normal way to modify the WebSphere Portal configuration is by editing the wpconfig.properties file and then running a `wpsconfig` task to load the configuration. At the time of writing, attempting to reconfigure WebSphere Portal for LDAP SSL connections using this approach resulted in a non-functional Portal engine and exceptions in WMM and ExternalAccessControl subsystems.
>
> As an alternative, we updated the wmm.xml file directly. Beware that `wpsconfig` tasks that involve LDAP configuration will overwrite this file, and you might need to edit it again after running `wpsconfig`.

Edit the C:\WebSphere\PortalServer\shared\app\wmm\wmm.xml file and modify the settings in the `<ldapRepository>` stanza for ldapPort and java.naming.security.protocol, as shown in Example 5-4 on page 154.

Example 5-4   Sample wmm.xml file

```
<ldapRepository name="wmmLDAP"
            UUID="LDAP1"

adapterClassName="com.ibm.ws.wmm.ldap.ibmdir.IBMDirectoryAdapterImpl"
            supportDynamicAttributes="false"

configurationFile="C:/ibm/WEBSPH~1/PORTAL~1/wmm/wmmLDAPServerAttributes.xml"
            wmmGenerateExtId="true"
            supportGetPersonByAccountName="true"
            profileRepositoryForGroups="LDAP1"
            supportTransactions="false"
            adminId="uid=wpsadmin,cn=users,o=ibm,c=us"
            adminPassword="5mApqlzuuPW7iNQpYWkEKQ=="
            ldapHost="phoenix.itsc.austin.ibm.com"
            ldapPort="636"
            ldapTimeOut="6000"
            ldapAuthentication="SIMPLE"
            ldapType="0"
            java.naming.security.protocol="ssl"
            groupCacheRefreshInterval="-1">
```

4. Restart the WebSphere_Portal application server.

5. Verify WebSphere Portal by logging in as the user wpsadmin (we used
   `https://pretoria.itsc.austin.ibm.com/portal/wps/myportal`).

5.5.3 Enabling SSL for Lotus Team Workplace

For the Lotus Team Workplace server to communicate with IBM Directory Server
over SSL, the following updates need to be made in the QuickPlace
administration place:

1. Using a browser, access the QuickPlace administration place (ours is in
   `https://kingston.itsc.austin.ibm.com/quickplace`).

2. Sign in as the QuickPlace Administrator, qpadmin.

3. Select **Server Setting** → **User Directory** → **Change Directory**.

4. Change the port number to 636.

5. Select the option for SSL connection with LDAP User Directory.

6. Click **Next** to save your changes.

7. Restart your Team Workplace server for the update to take effect.

5.5.4 Enabling SSL for Lotus Instant Messaging and Web Conferencing

For the Lotus Instant Messaging and Web Conferencing sever to communicate with IBM Directory Server over SSL, the following updates need to be made to the Directory Assistance document and Instant Messaging and Web Conferencing configuration database:

1. Change the Directory Assistance document:

   a. Select **File → Database → Open** to open the Directory Assistance database on the Instant Messaging and Web Conferencing server in a Notes client, and select the Directory Assistance database.

   b. Double-click the LDAP document to edit the Directory Assistance LDAP document.

   c. Click the **LDAP** tab.

   d. Change Channel Encryption to **SSL**.

   e. Save the LDAP document and close the Directory Assistance database.

2. Change the Instant Messaging and Web Conferencing configuration database (stconfig.nsf):

   a. Select **File → Database → Open** to open the stconfig.nsf database, and select the Instant Messaging and Web Conferencing configuration database.

   b. Double-click **LDAP Servers** in the left navigator pane.

   c. Change SSL Enabled to **True**.

   d. Save the LDAP Server document.

3. Create an Internet cross-certificate to IBM Directory Server:

   a. Copy the server.id file from the Lotus Instant Messaging and Web Conferencing server to the administrator's local machine. The server ID file is located in the Domino data directory (C:\Lotus\Domino\Data in our example).

   b. From the Domino administrator, select **File → Security → Switch ID**. Switch to the server.id file.

   c. Select **File → Security → User Security** and enter the password for the server ID if there is one.

   d. Select **Identity of Others → People, Services**.

   e. Select **Find out more about people/services** and then click **Retrieve Internet service certificate**.

f. Enter the host name of the server to be trusted (phoenix.itsc.austin.ibm.com in our example). Click **OK** to create the Internet cross-certificate.

g. Open the local Name and Address book (located in C:\Lotus\Notes\data\names.nsf in our example).

h. Select **Advanced** → **Certificates** → **Internet Cross Certificates**. Select the newly created cross-certificate and copy it to the Clipboard.

i. Select **File** → **Security** → **Switch ID**. Switch back to the administrator ID file (C:\Lotus\Notes\Data\user.id in our example).

j. Open the Domino Directory on the Instant Messaging and Web Conferencing server.

k. Select **Servers** → **Certificates** → **Internet Cross Certificates** and paste the cross certificate in the view.

4. Restart the Instant Messaging and Web Conferencing server for the changes to take effect.

5.6 SSL between the WebSphere Portal and Domino applications

There are three communication protocols between the WebSphere Portal Collaborative Services and back-end Lotus Domino servers, HTTP, DIIOP and LDAP. The HTTP connection also extends to the browser, but the DIIOP and LDAP connections are strictly between the WebSphere Portal server and Domino server.

> **Note:** Before completing the steps in this section, make sure that you have configured SSL on the desired ports as described in 5.2, "Enabling SSL on Domino-based products" on page 121.

To protect these connections, complete the sections for the back-end Domino Extended products you configured to only listen on SSL. We divide these sections into the following topics:

► Connecting the cs.jar file to the Domino mail and application servers over SSL

► Connecting cs.jar to Domino LDAP over SSL

► Configuring the Domino portlets for SSL connection

Depending on the portlets, the Collaborative Services can communicate with the Lotus Team Workplace server over HTTP and DIIOP, as well as the Domino LDAP server over LDAP. This section explains when each protocol is used and

how to reconfigure the Collaborative Services and each portlet to use SSL. We divide this section into the following topics:

► Connecting cs.jar to Lotus Team Workplaces over SSL
► Configuring the Team Workplace portlets to connect over SSL

The Lotus Instant Messaging and Web Conferencing portlets connect to the Instant Messaging and Web Conferencing portlet over HTTP. The following sections explain what changes need to be made so that the collaborative services will use HTTPS and what changes need to be made to each portlet to connect to HTTPS:

► Connecting cs.jar to the Instant Messaging and Web Conferencing server over SSL
► Configuring Instant Messaging and Web Conferencing portlets to connect over SSL

5.6.1 Connecting the cs.jar file to the Domino mail and application servers over SSL

The Collaborative Services running on the WebSphere Portal server connect your Domino mail and application servers over the HTTP and DIIOP protocols.

To connect the cs.jar file to the Domino servers you need to go through several steps. For the handshake to occur between the Collaborative Services and Domino servers for all Domino portlets, they first need to exchange certificates. Except for Domino Web Access portlet HTTP access, these steps are necessary for communication with Domino Server through SSL.

To connect the cs.jar file to the Domino mail and application servers over SSL, complete the following steps:

1. Extract the Domino SSL key:

   a. Copy the key ring file and stash file from the Domino server to the WebSphere Portal server. These files are generated for each Domino server, as discussed in 5.2, "Enabling SSL on Domino-based products" on page 121.

   b. Select **Programs** → **IBM HTTP Server** → **Start Key Management Utility** to open the iKeyman utility shipped with IBM HTTP server.

   c. Select **Key Database File** → **Open** to open the Domino key file and enter the password.

   d. Select **Personal Certificates** from the drop-down list, as shown in our example in Figure 5-18 on page 158.
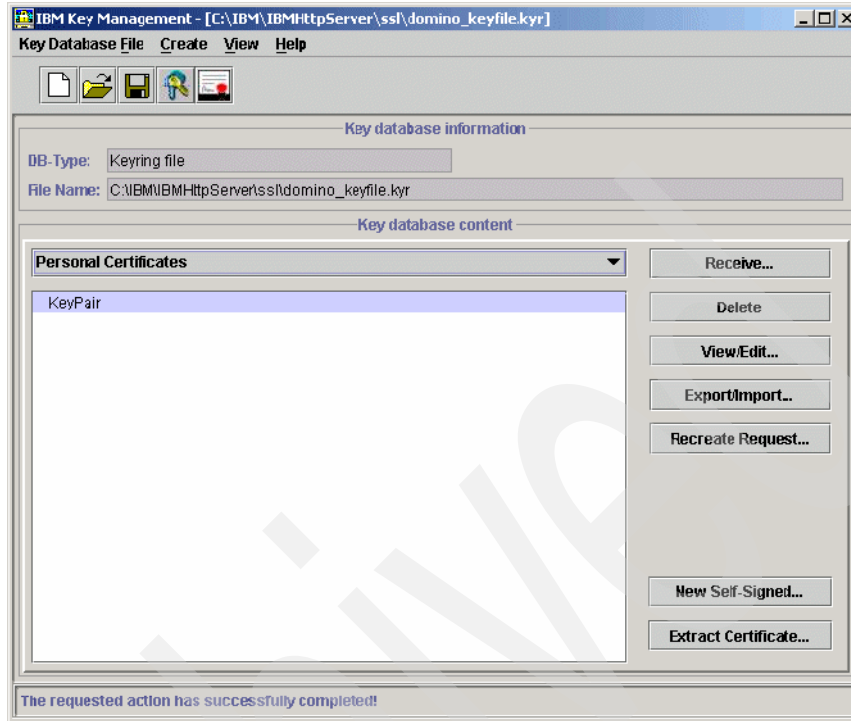
Figure 5-18   HTTP iKeyman: Personal Certificates

    e. Click **Export/Import**. Select the **Export** option with key file type PKCS12. We used the name dominocert.p12. Save the file, click **OK**, and supply a password to protect this file.

    f. Close the iKeyman utility.

2. Import the Domino key into WebSphere server trust file:

    a. Run the iKeyman tool from C:\WebSphere\Appserver\bin. Issue the `ikeyman.bat` command.

    b. Select **Key Database File** → **Open** to open the default server trust file from C:/WebSphere/AppServer/etc/DummyServerTrustFile.jks. Supply the password `WebAS`.

    c. Go to **Personal Certificates** and click **Import**. Proceed to import the certificate file that was extracted from Domino key file store into the trust file. The Domino exported key file is in the PKCS12 format. You should see KeyPair in the Personal Certificates.

> **Note:** The Domino server personal certificate will always import with the name keyfile. If you import another key file from a Domino server, you will be prompted to enter a new name; any name is fine in this field.

3. Import the Domino key into WebSphere client trust file. Perform the same steps as for the server trust file, but use the file C:/WebSphere/AppServer/etc/DummyClientTrustFile.jks.

4. Configure the Collaborative Services to connect to DIIOP over SSL.

   For the handshake to occur between the Collaborative Services running on WebSphere Portal, and the Domino DIIOP task, all of the Domino servers' SSL certificates must be signed by a single certificate authority (CA) (in our example, Domino is the single CA). Then, you must copy the TrustedCerts.class file from a Domino server running the DIIOP task to the WebSphere Portal server. The TrustedCerts.class file does not get created until the DIIOP task loads on SSL. The TrustedCerts.class file is stored within the <domino data>\domino\java directory to the WebSphere Portal server machine's class directory <wp_root>\shared\app\.

5. Restart WebSphere Portal application server.

5.6.2 Connecting cs.jar to Domino LDAP over SSL

For the handshake to occur between the Collaborative Services running on WebSphere Portal and the Domino LDAP task, complete the following steps:

1. Exchange the Domino SSL certificate with WebSphere Application Server.

   5.6.1, "Connecting the cs.jar file to the Domino mail and application servers over SSL" on page 157 describes these steps in detail. The steps are:

   a. Extract the Domino key.

   b. Import the key to the WebSphere server trust file DummyServerTrustFile.jks.

   c. Import the key to the WebSphere client trust file DummyClientTrustFile.jks.

2. Configure CSEnvironment.properties for SSL. The following changes allow a connection to LDAP using SSL:

   ```
   CS_SERVER_DOMINO_DIRECTORY_1.port=636
   CS_SERVER_DOMINO_DIRECTORY_1.ssl=true
   ```

3. Restart WebSphere Portal for the changes to take effect.

5.6.3 Configuring the Domino portlets for SSL connection

The Domino portlets, Domino Web Access (formerly called iNotes), and Notes and Domino attempt to connect to the Domino resources over HTTP by default. In this section, we explain how you can edit the portlets to use HTTPS.

Domino Web Access

Users have edit access to this portlet by default if they have edited this portlet to change the default frame height or width. The user will need to edit the portlet again and change the radio button from http to https or detect protocol automatically. If the automatically detect mail file option is not configured to work, the detect protocol automatically option will not work either.

For users who have not edited the portlet, the portal administrator can sign on, edit the portlet, select either the https or detect protocol automatically, and save the change. This change will now be the default for the portlet.

Notes and Domino

As the Portal administrator, sign on and browse to any Notes and Domino portlet you have deployed. Edit the portlet, and edit any view you have created. The next window opens with the Notes pickers. At the bottom of this window, you have the option for http, https, or detect protocol automatically. Select either https or detect protocol automatically. If you configured the LDAP server and the bind user with the Domino LDAP task, the detect protocol automatically option will not work.

5.6.4 Connecting cs.jar to Lotus Team Workplaces over SSL

Regardless of the portlet, a connection will be made from the Collaborative Services and the user's browser directly to the Team Workplace server. If you want this connection to occur over SSL, complete the following steps:

1. Exchange the Domino SSL certificate with WebSphere Application Server.

   5.6.1, "Connecting the cs.jar file to the Domino mail and application servers over SSL" on page 157 describes these steps in detail. The steps are:

   a. Extract the Domino key.

   b. Import the key to the WebSphere server trust file DummyServerTrustFile.jks.

   c. Import the key to the WebSphere client trust file DummyClientTrustFile.jks.

2. Configure CSEnvironment.properties for SSL. The following changes allow an SSL connection to a Team Workplaces server:

```
CS_SERVER_QUICKPLACE_1.protocol=https
CS_SERVER_QUICKPLACE_1.port=443
```

3. Restart WebSphere Portal for the changes to take effect.

5.6.5 Configuring the Team Workplace portlets to connect over SSL

The Lotus Team Workplace portlets, consisting of Lotus QuickPlace, QuickPlace Inline, and My Team Workplaces, attempt to connect to the Domino resources over HTTP. This section explains how you can edit the portlets to use HTTPS. Also, some of portlets use the Domino connections from CSEnvironment.properties and need to be configured to use SSL as well.

Lotus QuickPlace portlet

The following was taken from the *IBM WebSphere Portal 5.0.2.2 Release Notes*:

> The Lotus QuickPlace portlet that was released originally as part of the WebSphere Portal 4.2 Enable offering (deployed from the quickplace.war file) does not support the SSL security portal protocol.

Based on this quote, if your organization must use SSL, remove the Lotus QuickPlace portlet from the portal and deploy the Inline QuickPlace portlet from the quickplace2.war file instead.

QuickPlace Inline portlet

To change the QuickPlace Inline portlet to connect to the Team Workplace server over HTTPS, complete the following steps:

1. Sign in to WebSphere Portal as the Portal administrator.
2. Select **Administration** → **Portlets** → **Manage Portlets**.
3. Scroll down to the QuickPlace Inline portlet, and click **Modify Parameters**.
4. Change defaultURL to use HTTPS; set useSSL to **Yes**.
5. Click **Save**, and then **Cancel** to exit out of the portlet parameters.

For users who can put the portlet in edit mode, the server picker and database picker can also make connections to Domino servers over SSL. Refer to 5.6.2, "Connecting cs.jar to Domino LDAP over SSL" on page 159 for the LDAP connection for server picker. Refer to 5.6.1, "Connecting the cs.jar file to the Domino mail and application servers over SSL" on page 157 for database picker processing.

My Team Workplaces portlet

The My Team Workplaces portlet connects to the Team Workplace server over both HTTP and DIIOP. In this section, we explain how to:

► Configure the My Team Workplaces portlet to connect over HTTPS

- ► Ensure that My Places works over SSL
- ► Ensure that Search all places works over SSL

Configuring the My Team Workplaces portlet to connect over HTTPS

By default, the My Team Workplaces portlet will connect to the Team Workplace portlet over HTTP. To direct it to use HTTPS, complete the following steps:

1. Sign in to WebSphere Portal as the Portal administrator.

2. Select **Administration** → **Portlets** → **Manage Portlets**.

3. Scroll down to the My Lotus Team Workplaces portlet, and click **Modify Parameters**.

4. Set QuickPlacePort to `443` and QuickPlaceProtocal to `https`.

5. Click **Save**, and then **Cancel** to return to the Manage Portlets page.

Ensuring that My Places works over SSL

If the Open workplace link continues to direct you to HTTP, and not HTTPS, make sure that you completed 5.2.3, "Enabling SSL on Lotus Team Workplace" on page 137 for correcting the connection to My Places.

Ensuring that Search all places works over SSL

If you enabled SSL on the Team Workplace Domain Catalog server, the Search workplaces, Search this workplace, My Pages, and My Tasks might stop working. If this is the case, make sure you completed 5.2.3, "Enabling SSL on Lotus Team Workplace" on page 137.

5.6.6 Connecting cs.jar to the Instant Messaging and Web Conferencing server over SSL

Updating the Collaborative Services to point to the Instant Messaging and Web Conferencing server over HTTPS results in the stlinks applet, loaded into your browser and responsible for awareness in WebSphere Portal, to connect to the Instant Messaging and Web Conferencing server over HTTPS. To do this, complete the following steps:

1. Exchange the Domino SSL certificate with WebSphere Application Server.

   5.6.1, "Connecting the cs.jar file to the Domino mail and application servers over SSL" on page 157 describes these steps in detail. The steps are:

   a. Extract the Domino key.

   b. Import the key to the WebSphere server trust file DummyServerTrustFile.jks.

   c. Import the key to the WebSphere client trust file DummyClientTrustFile.jks.

2. Configure CSEnvironment.properties for SSL. The following changes allow the connection to the Instant Messaging and Web Conferencing server using SSL:

```
CS_SERVER_SAMETIME_1.protocol=https
CS_SERVER_SAMETIME_1.port=443
```

3. Restart WebSphere Portal for the changes to take effect.

5.6.7 Configuring Instant Messaging and Web Conferencing portlets to connect over SSL

The portlets that need to be updated to point to the Instant Messaging and Web Conferencing portlet over SSL are the Sametime Connect and My Lotus Web Conferencing portlets. The Who Is Here and Sametime Contact List portlets will connect over HTTPS as soon as the updates have been made to CSEnvironment.properties.

Sametime Connect portlet

To configure the Sametime Connect portlet, complete the following steps:

1. Sign in to WebSphere Portal as the Portal administrator.

2. Go to the Sametime Connect portlet.

3. Click the pencil icon to put the portlet into edit mode.

4. Change the server name to use HTTPS protocol (in our example, `https://laredo.itsc.austin.ibm.com`) and change the port to 443.

5. Click **Save** to save the setting and return to the Manage Portlets page.

My Lotus Web Conferencing portlet

The My Lotus Web Conferencing portlet will initiate a connection between the Collaborative Services and the Instant Messaging and Web Conferencing server. For the handshake between these to occur, complete the following steps:

1. Exchange the Domino SSL certificate with WebSphere Application Server. 5.6.1, "Connecting the cs.jar file to the Domino mail and application servers over SSL" on page 157 describes these steps in detail. The steps are:

   a. Extract the Domino key.

   b. Import the key to the WebSphere server trust file DummyServerTrustFile.jks.

   c. Import the key to the WebSphere client trust file DummyClientTrustFile.jks.

2. Update the portlet.

   After you have exchanged the personal certificates, you need to configure the portlet to connect to the Instant Messaging and Web Conferencing server over SSL by completing the following steps:

   a. Sign in to WebSphere Portal as the Portal administrator.

   b. Select **Administration** → **Portlets** → **Manage Portlets**.

   c. Scroll down to the Lotus Web Conferencing portlet, and click **Modify Parameters**.

   d. Change SametimePort1 to `443` and SametimeUseSSL1 to **Yes**.

   e. Click **Save** to save the settings, and then **Cancel** to return to the Manage Portlets page.

At this point, the Web Conferencing portlet will connect to the Instant Messaging and Web Conferencing server over SSL.

5.7 SSL between Team Workplace and Instant Messaging and Web Conferencing

If you configured Lotus Team Workplace for Sametime awareness and chat or Sametime meetings, you now need to configure these features over SSL.

5.7.1 Configuring Instant Messaging (Sametime) awareness and chat over SSL

To configure Team Workplace to connect to the Instant Messaging server for online awareness and chat over SSL, you need to update the Community server field in the Team Workplace administration console:

1. With a Web browser, go to the Team Workplace server administration console (in our case, `https://kingston.itsc.austin.ibm.com/quickplace`). Sign in as the server administrator, qpadmin.

2. Select **Server Settings** → **Other Options** → **Edit Options**.

3. Under the Sametime Servers heading, change the Sametime Instant Messaging server in the community field URL from http to https (in our example, `https://laredo.itsc.austin.ibm.com`).

4. Click **Next**, and then sign out of Team Workplace.

5.7.2 Configuring Web Conferencing (Sametime) meetings over SSL

To configure the Team Workplace server to connect to the Web Conferencing server over SSL, complete the following steps:

1. Update the Sametime Meeting server field in the Team Workplace administration console.

   a. Sign in as the administrator, qpadmin, from the Team Workplace server administration console (in our example, `https://kingston.itsc.austin.ibm.com/quickplace`).

   b. Select **Server Settings → Other Options → Edit Options**.

   c. Under Sametime Servers, type the full URL for the HTTPS of the Sametime Meeting server (in our example, `https://laredo.itsc.austin.ibm.com`).

   d. Click **Next**, and then sign out of Team Workplace.

2. Set up the SSL handshake between Team Workplace and Web Conferencing.

   a. Copy the Instant Messaging and Web Conferencing servlets key file (stkeys.pfx in our example) from the Instant Messaging and Web Conferencing server's program directory to the Team Workplace server's program directory (C:\Lotus\Domino on both server in our environment).

   b. Open a text editor on the Team Workplace server and create a text file consisting of the following entries, matching the one from our example in Example 5-5 on page 166:

   - VPS_NAME: The Domino canonical name of your Instant Messaging and Web Conferencing server.

   - SametimeCluster: The Domino canonical name of your Instant Messaging and Web Conferencing server.

   - ConfigurationHost: The host name of your Instant Messaging and Web Conferencing server.

   - ConfigurationPort: The HTTP port on which your Instant Messaging and Web Conferencing server are listening.

   - ConfigurationSSLEnabled: This should be set to **True**.

   - SSLManagerClassName: This line is case sensitive, and it is very important that you type exactly the following:
     `com.lotus.sametime.configuration.IBMJSSE118Manager`

   - javax.net.ssl.keyStore: The path to the Instant Messaging and Web Conferencing servlets key file on the Team Workplace server.

   - javax.net.ssl.trustStore: The path to the Instant Messaging and Web Conferencing servlets key file on the Team Workplace server.

- javax.net.ssl.keyStorePassword: The password for the Instant Messaging and Web Conferencing servlets key on the Team Workplace server.

- javax.net.ssl.trustStorePassword: The password for the Instant Messaging and Web Conferencing servlets key on the Team Workplace server.

- SameTimeAdminUsername: This should be the same user specified in the qpconfig.xml file configured in "Specifying the Web Conferencing authentication name" on page 113.

- SameTimeAdminPassword: The password for the user specified earlier in "Specifying the Web Conferencing authentication name" on page 113.

Example 5-5   Sample of sametime.ini on Team Workplace server

```
[Config]
VPS_NAME=CN=laredo/O=itso
SametimeCluster=CN=laredo/O=itso
ConfigurationHost=laredo.itsc.austin.ibm.com
ConfigurationPort=443
ConfigurationSSLEnabled=true
SSLManagerClassName=com.lotus.sametime.configuration.IBMJSSE118Manager
javax.net.ssl.keyStore=c:\lotus\domino\stkeys.pfx
javax.net.ssl.trustStore=c:\lotus\domino\stkeys.pfx
javax.net.ssl.keyStorePassword=passw0rd
javax.net.ssl.trustStorePassword=passw0rd
SameTimeAdminUsername=CN=Domino Admin/O=itso
SameTimeAdminPassword=passw0rd
```

c. Save this file as sametime.ini in the Team Workplace's Domino program directory.

d. Restart the Team Workplace server for the new setting to load.

6

Incorporating IBM Tivoli Access Manager for e-business

In this section, we discuss gaining additional protection using IBM Tivoli Access Manager for e-business as the front-end authentication and authorization agent. We base this discussion on the IBM Redbook *Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1*, SG24-6325. We discuss the following topics:

► Overview

► Installing the policy server node

► Installing the reverse proxy node

► Java Runtime Environment on WebSphere Portal

► Enabling SSL between WebSEAL and WebSphere Portal

► Configuring WebSphere Portal for access authorization

► Configuring WebSphere Portal authentication

► Protecting Domino Extended Products

167

6.1 Overview

IBM Tivoli Access Manager for e-business provides a means to secure Web services from an authorization and authentication point of view. This solution provides its function using the structure shown in Figure 6-1.
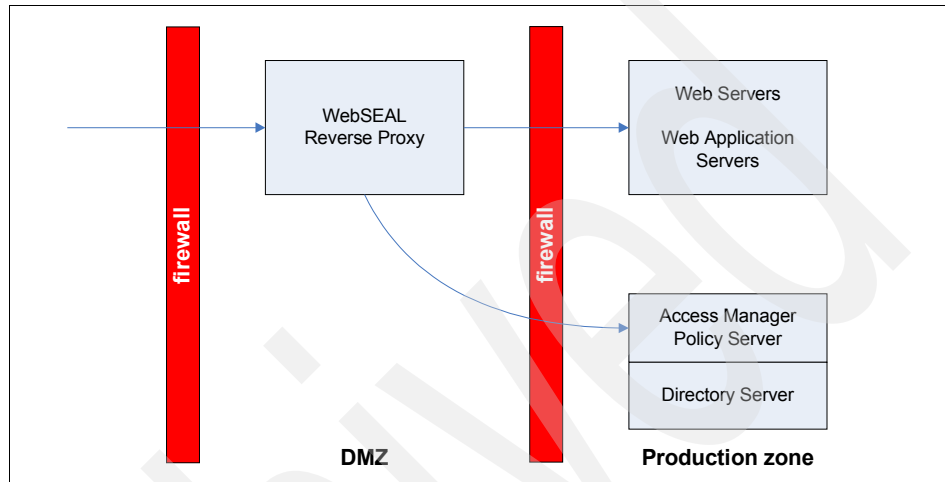
Figure 6-1 Structure of a secured Web service

In Figure 6-1, IBM Tivoli Access Manager for e-business consists of two servers:

► The reverse proxy node, which resides in the demilitarized zone (DMZ), is the primary interface for user access. It translates all the Web access requests into an internal request and communication stream to a back-end Web server or Web application server.

► The policy server node, which resides in the secured management area, provides an authentication and authorization mechanism as requested by the reverse proxy node.

There are two options for how to connect the back-end Web services to the reverse proxy engine. The options depend on how security authentication information in being passed. You can protect WebSphere Portal with a Trust Association Interceptor (TAI) junction or Lightweight Third Party Authentication (LTPA) junction. The TAI junction will provide more flexibility but a little less security than the LTPA junction:

► Flexibility of the TAI junction

The TAI junction cannot be used with Domino applications. Therefore, Domino applications can only be connected to Tivoli Access Manager using an LTPA junction.

► Security of the LTPA junction

With an LTPA junction to WebSphere Portal, you can also create LTPA junctions to the back-end Domino products, and the entire environment is protected by IBM Tivoli Access Manager.

Before deciding what type of junction you will deploy, install the IBM Tivoli Access Manager policy server node, as discussed in 6.2, "Installing the policy server node" on page 169, and the WebSEAL reverse proxy server node, as discussed in 6.3, "Installing the reverse proxy node" on page 176.

After the IBM Tivoli Access Manager servers are installed, decide what junction you need in your environment and complete the steps in the following sections:

► Java Runtime Environment on WebSphere Portal

► Enabling SSL between WebSEAL and WebSphere Portal

► Configuring WebSphere Portal for access authorization

► Configuring WebSphere Portal authentication

► Protecting Domino Extended Products

In this section, we discuss the installation of IBM Tivoli Access Manager for e-business. For the complete discussion, see *Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1*, SG24-6325. The overview we discuss here is the procedure that we used to set up our environment. We divide this discussing into the policy server installation and the reverse proxy server installation.

> **Note:** When installing and configuring the policy server node, we referenced the following product guides and Redbooks:
>
> ► *IBM Tivoli Access Manager Base Installation Guide, V5.1*, SC32-1362
>
> ► *IBM Tivoli Access Manager Base Administration Guide, V5.1*, SC32-1360
>
> ► *Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1*, SG24-6325
>
> ► *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014

6.2 Installing the policy server node

This section describes how to install and configure IBM Tivoli Access Manager on IBM Tivoli Directory Server.

> **Note:** This section assumes that you have installed Tivoli Directory Server with all the updates from Chapter 7, "Integrating directory servers in an IBM WebSphere Portal environment" on page 221.

The high-level tasks to install the policy server node are as follows:

► Configuring Tivoli Directory Server for Tivoli Access Manager

► Installing Tivoli Access Manager

► Configuring Tivoli Access Manager

► Installing Tivoli Access Manager V5.1 Base Fix Pack 2

6.2.1 Configuring Tivoli Directory Server for Tivoli Access Manager

To use Tivoli Directory Server for IBM Tivoli Access Manager for e-business, complete the following steps:

1. Prepare the schema definitions. The additional schema definitions for IBM Tivoli Access Manager are added automatically during the installation of Tivoli Directory Server Version 5.2. If you are using Tivoli Directory Server V4.1 or V5.1, refer to the *IBM Tivoli Access Manager Base Installation Guide, V5.1*, SC32-1362, for additional steps for preparing the schema.

2. Create a suffix for the Tivoli Access Manager metadata:

   To create a suffix from the Tivoli Directory Server - Web Administration Tool for the Tivoli Access Manager metadata, complete the following steps:

   a. From the Tivoli Directory Server Web Administration Tool, select **Server administration** → **Manage server properties**.

   b. On the Manage server properties window, click **Suffixes**.

   c. Enter the Suffix DN `secAuthority=Default` and click **Add**.

   d. Click **OK** at the bottom of the page to save the settings.

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

6.2.2 Installing Tivoli Access Manager

This section describes how to install and configure IBM Tivoli Access Manager Version 5.1 on the policy server node. Tivoli Access Manager V5.1 can be installed using the installation wizard or native installation utilities by platform. We

choose to install it using the wizard. To install the IBM Tivoli Access Manager V5.1 Policy Server, complete the following steps:

1. Ensure that the Tivoli Directory Server, IBM GSKit 7.0.1.16, and Tivoli Directory Client SDK are installed and active.

2. Use the *Tivoli Access Manager Base for Windows NT, Windows XP, Windows 2000 and Windows 2003* CD, and run `Setup.exe` from the \Windows\PolicyDirector\Disk Images\Disk1 folder to start the installer.

3. When the Choose Setup Language window opens, select the desired language and click **OK**.

4. When the Welcome window opens, click **Next**.

5. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

6. When the Select Packages window opens, we selected the following packages and then clicked **Next**:
   – Access Manager Runtime
   – Access Manager Policy Server
   – Access Manager Authorization Server

7. When the Choose Destination Directory window opens, we entered the installation directory C:\IBM\Tivoli\tam and then clicked **Next**.

8. When the Summary/Start Copying Files window opens, review the selections and then click **Next**.

9. When the installation is complete, select **Yes, I will restart my computer now** and then click **OK**.

Note: This note describes how to manually stop the server prior to system restart. On occasion, if you do not stop the WebSphere Application Server from the command line by entering the following, you might not be able to start the WebSphere Application Server after the restart:

```
c:/ibm/WebSphere/AppServer/bin/stopServer server1
```

To resolve this issue, delete the file server.pid in the <was_home>/logs/server1 directory, and then restart the server.

6.2.3 Configuring Tivoli Access Manager

This section describes how to configure the following IBM Tivoli Access Manager components:

► Configuring Tivoli Access Manager runtime
► Configuring Tivoli Access Manager authorization server

- ► Configuring Tivoli Access Manager authorization server
- ► Updating the Windows registry for Tivoli Access Manager services
- ► Tivoli Access Manager Windows services startup

Configuring Tivoli Access Manager runtime

To configure the Tivoli Access Manager runtime, complete the following steps:

1. Ensure that the IBM Tivoli Directory Server is started.

2. Start the Tivoli Access Manager configuration by clicking **Programs** → **IBM Tivoli Access Manager** → **Configuration**, or run the `pdconfig` command.

3. From the Access Manager Configuration window, select **Access Manager Runtime** and click **Configure**.

4. Select **LDAP** and click **Next**.

5. When the LDAP Server Information window opens, we entered the host name and port of the Tivoli Directory Server and clicked **Next**.

6. When the SSL with the registry server window opens, select the appropriate setting about whether you have activate SSL and then click **Next**.

7. When the Logging Information window opens, we enabled Tivoli common directory logging into C:\IBM\Tivoli\common and then clicked **Next**.

8. When the Configuration Summary window opens, review the settings and click **Finish**.

The Tivoli Access Manager Configuration tool should display the configured status `Yes` for Tivoli Access Manager runtime.

Configuring the Tivoli Access Manager policy server

To configure the Tivoli Access Manager policy server, complete the following steps:

1. From the Access Manager Configuration window, select **Access Manager Policy Server** and click **Configure**.

2. When the LDAP Administration ID window opens, we entered the administrator of `cn=root` and its password, and then clicked **OK**.

3. When the Tivoli Access Manager administrator ID window opens, we entered the ID of `sec_master` and assigned a password, and then clicked **OK**.

4. When the Access Manager Policy Server SSL parameters window opens, we accepted the default at this time, and then clicked **OK**.

5. After the configuration is complete, you should see a message similar to the one shown in Figure 6-2 on page 173. Click **OK**.
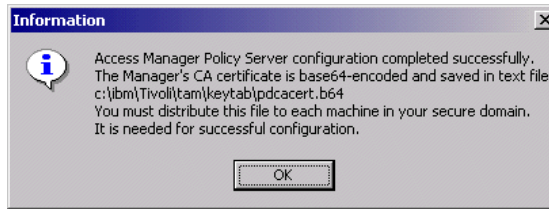
Figure 6-2   Tivoli Access Manager Policy Server configuration success

> **Tip:** The first time we configured our environment, we received an error
> message. We then reviewed the c:\ibm\Tivoli\tam\log\msg__config.log file. We
> found the error:
>
>   Error code 0x20 was received from the LDAP server. Error text: No such
>   object.
>
> We later discovered that the suffix for the Tivoli Access Manager metadata
> (secAuthority=Default) was not saved. We have included a note in the section
> where the suffix is added to ensure that you scroll down the page and click **OK**
> to save the suffix. After successfully adding the suffix, we reran the policy
> server configuration successfully.

The Tivoli Access Manager Configuration tool should display the configured
status `Yes` for the Access Manager policy server.

Configuring Tivoli Access Manager authorization server

To configure the Tivoli Access Manager authorization server, complete the
following steps:

1. From the Access Manager Configuration window, select **Access Manager
   Authorization Server** and click **Configure**.

2. When the Domain Information window opens, enter `Default` in the Domain
   field and click **Next**.

3. When the Policy Server Information window opens, we entered the policy
   server host name `phoenix` and port `7135`, and then clicked **Next**.

4. When the Administrator ID for domain Default window opens, we used
   `sec_master` and its password, and then clicked **Next**.

5. When the Authorization Server window opens, we entered the host name
   `phoenix`, administration port `7137`, authorization port `7136`, and then clicked
   OK.

The Access Manager Configuration tool should display the configured status Yes for the Access Manager authorization server. You can close the Access Manager Configuration utility by clicking **Close**.

Updating the Windows registry for Tivoli Access Manager services

While writing this book, we found that Tivoli Access Manager V5.1, Access Manager policy server Windows services did not start properly. We discovered that the Windows registry for the Access Manager policy server Windows service was not configured correctly by the Tivoli Access Manager V5.1 installation program. This is caused by a non-standard installation path.

To work around this configuration issue, we did the following:

1. Start the Registry Editor, using the command `regedit`.

2. To update the Access Manager policy server Windows service definition in the registry, update the key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IVMgr, edit the key ImagePath, and provide the value of the appropriate path for pdmgrd.exe.

3. To update the Access Manager authorization server Windows service definition in the registry, update the key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IVACId, edit the key ImagePath, and provide the appropriate path for pdacld.exe.

4. Close the Registry Editor.

Tivoli Access Manager Windows services startup

In our example, we installed Tivoli Directory Server and the Tivoli Access Manager components on the same node (the policy server node). The Tivoli Access Manager Access Manager Auto-Start Service is set to automatic startup by default. The purpose of the Access Manager Auto-Start Service is to start other Tivoli Access Manager services automatically, such as the policy server and authorization server.

The Tivoli Access Manager services startup is dependent on the Tivoli Directory Server being started. Even after attempting to make the Access Manager Auto-Start Service dependent on the Tivoli Directory Server service startup, we found that it timed out. To resolve this issue, we set the Access Manager Auto-Start Service to manual, and manually started the service after the Tivoli Directory Server service had completed startup.

6.2.4 Installing Tivoli Access Manager V5.1 Base Fix Pack 2

To install Tivoli Access Manager V5.1 Base Fix Pack 2, complete the following steps.

> **Note:** For more detailed information about installing the Tivoli Access Manager V5.1 Base Fix Pack 2, refer to the *readme*, available at:
>
> ```
> http://www.ibm.com/support/docview.wss?rs=203&context=SW000&q1=%2bfix+%2bTiv
> oli+%2bAccess+%2bManager&uid=swg24006925&loc=en_US&cs=utf-8&cc=us&lang=all
> ```

1. Back up the Tivoli Access Manager V5.1 Base databases on the policy server node before installing Fix Pack 2 as follows:

    a. Create a database backup directory (for example, c:\ibm\tamdb.bak).

    b. Open a command window and enter the following command:

    ```
    pdbackup -action backup -list c:\ibm\Tivoli\tam\etc\pdbackup.lst -path
    c:\ibm\tamdb.bak
    ```

2. Ensure that you have installed IBM GSKit V7.0.1.16, which is a prerequisite to Fix Pack 2. Refer to "Installing IBM GSKit V7.0.1.16" on page 224 for details.

3. Ensure that the following Tivoli Access Manager Windows services are stopped before installing the fix pack:

    – Tivoli Access Manager authorization server
    – Tivoli Access Manager policy server

4. If you have Web Portal Manager, ensure that the server1 application server is stopped.

5. Download Tivoli Access Manager V5.1 Base Fix Pack 2 from the following URL to a temporary directory (for example, c:\temp\tam51.fp2) on the policy server node:

    ```
    http://www.ibm.com/support/docview.wss?rs=203&context=SW000&q1=%2bfix+%2bTi
    voli+%2bAccess+%2bManager&uid=swg24006925&loc=en_US&cs=utf-8&cc=us&lang=all
    ```

6. Navigate to the temporary directory where you downloaded the fix pack (for example, c:\temp\tam51.fp2), and run 5.1-TAM-FP02-WIN.exe to start the Access Manager V5.1 Fix Pack Setup (5.1.0.2).

7. When the Welcome window opens, click **Next**.

8. When the License Agreement window opens, review the terms, and if in agreement, click **Yes**.

9. When the installation is complete, click **Finish**.

10. When the Configuration Type window opens, select **Full** and click **Next**.

11. When the Java Runtime Environment window opens, specify the path to the JRE and then click **Next**.

> **Note:** In our example, the PDJRTE was already configured prior to starting the Fix Pack 2 installation. When the dialog box opens to configure the PDJRTE, it gave us an error dialog box stating that it was already configured. We clicked **OK** and continued. Then, we clicked **Cancel**.
>
> The fix pack installer on Windows launches the PDJRTE configuration automatically as part of the fix pack installation even though it is configured in our example.

12. To verify that the fix pack installation was successful, enter the command `pdversion`. This should return a list of components installed at the 5.1.0.2 level (Fix Pack 2).

13. Restart the following Access Manager Auto-Start Service Windows services:
    – Tivoli Access Manager authorization server
    – Tivoli Access Manager policy server

The installation and base configuration for the policy server node components are complete.

6.3 Installing the reverse proxy node

This section describes the procedure we used to install and configure the reverse proxy node for our sample environment on Microsoft Windows.

> **Note:** When installing and configuring the reverse proxy node, we referenced the following product guides and redbook:
>
> ► *IBM Tivoli Access Manager for e-business Web Security Installation Guide, V5.1*, SC32-1361
>
> ► *IBM Tivoli Access Manager WebSEAL Administration Guide 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 are as follows:

► Prerequisites

► Tivoli Access Manager: Installing WebSEAL

- ▶ Tivoli Access Manager: Configuring WebSEAL
- ▶ Installing Tivoli Access Manager V5.1 Base Fix Pack 2
- ▶ Installing Tivoli Access Manager V5.1 WebSEAL Fix Pack 2

6.3.1 Prerequisites

Before setting up the reverse proxy node, you need to ensure that the following are ready in the machine:

- ▶ IBM GSKit

  In our example, we installed the IBM GSKit V7.0.1.16 on the reverse proxy node. In this case, the Windows registry entry for the application name is `policydirector`.

- ▶ Java Runtime Environment (JRE)

  The Java Runtime Environment is needed by the iKeyman utility installed with the GSKit. The iKeyman utility is used to create and manage the keystore and certificates.

- ▶ Join the Active Directory domain

  If you are installing Tivoli Access Manager with Active Directory, you need to join the Active Directory domain.

6.3.2 Tivoli Access Manager: Installing WebSEAL

The Tivoli Access Manager V5.1 WebSEAL can set up this system using one of the following installation methods:

- ▶ Installation wizard
- ▶ Native installation utilities

Installing IBM Tivoli Directory Client

To install the Tivoli Directory Client Version 5.2 on the reverse proxy node, complete the following steps:

1. Insert the *IBM Tivoli Access Manager Web Security for Windows 2000* CD.

2. Navigate to the <CD_Root>\windows\Directory folder and run `Setup.exe` to start the installation program.

3. When the Select Language used by install Wizard window opens, select the desired language and click **OK**.

4. When the Welcome window opens, click **Next**.

5. When the License Agreement window opens, review the terms, and if in agreement, select **I accept the terms in the license agreement**, and then click **Next**.

6. The installer will detect applications that have already been installed (for example, GSKit). Click **Next**.

7. When the Tivoli Directory Server V5.2 Installation Directory window opens, we used C:\IBM\LDAP and then clicked **Next**.

8. Select the Tivoli Directory Server language (for example, English) and click **Next**.

9. When the Select Features to install window opens, we selected the following values and then clicked **Next**:

   – Select **Client SDK 5.2**.
   – Clear GSKit, because a later level of the GSKit has already been installed.

10. When the Summary window opens, review the selections and click **Next**.

11. When the installation completes, review the *readme* files for the client and click **Next**.

12. When prompted, select **Yes, restart my computer** and click **Next**.

13. Click **Finish**.

Installing the Tivoli Access Manager WebSEAL

To install and configure a IBM Tivoli Access Manager WebSEAL server system on Windows, complete the following steps:

1. Log on as a user with administrator privileges and ensure that the following prerequisites have been met.

   – Ensure that Tivoli Directory Server and Tivoli Access Manager policy server are up and running in normal mode.
   – Ensure that the GSKit and Java Runtime Environment are installed.
   – Ensure that the IBM Tivoli Directory Client is installed.

2. Insert the *IBM Tivoli Access Manager Web Security for Windows 2000* CD.

3. Navigate to the <CD_Root>\windows\PolicyDirector\Disk Images\Disk1 folder and run `Setup.exe` to start the installation program.

4. When the Choose Setup Language window opens, select the desired language (for example, English) and click **OK**.

5. When the Welcome window opens, click **Next**.

6. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

7. When the Select Packages window opens, we selected the following packages and then clicked **Next**:
   – Access Manager Runtime
   – Access Manager Java Runtime Environment
   – Access Manager Web Security Runtime
   – Access Manager WebSEAL

8. Install Tivoli Access Manager runtime:

   a. When the Access Manager Runtime Choose Destination Directory window opens, we selected C:\ibm\Tivoli\tam and click **Next**.

   b. When the Access Manager Runtime Installation Summary window opens, click **Next** to begin copying files.

9. Install Tivoli Access Manager Web Security runtime:

   a. When the Welcome Access Manager Web Security Runtime window opens, click **Next**.

   b. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

   c. When the Access Manager Java Runtime Environment Choose Destination Directory window opens, we selected c:\ibm\Tivoli\PDWebRTE and then clicked **Next**.

   d. When the installation is complete, select **Yes, I want to restart my computer now**. Click **Finish**.

   Note: Even though we selected **Yes, I want to restart my computer now** and clicked **Finish**, the installation program will continue to the Access Manager WebSEAL installer.

10. Install Tivoli Access Manager WebSEAL:

    a. When the Welcome Access Manager WebSEAL window opens, click **Next**.

    b. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

    c. When the Access Manager WebSEAL Choose Destination Directory window opens, we used c:\ibm\Tivoli\PDWeb and then clicked **Next**.

    d. When the installation is complete, select **Yes, I want to restart my computer now**. Click **OK**. This time, the system will restart.

6.3.3 Tivoli Access Manager: Configuring WebSEAL

After the Tivoli Access Manager V5.1 WebSEAL and Web Security runtime are installed, they must be configured.

To configure the Tivoli Access Manager runtime and WebSEAL, complete the following steps:

1. Start the `pdconfig` configuration utility on the reverse proxy node.

2. From the Access Manager Configuration utility, select the Access Manager Runtime package and click **Configure**:

   a. When the Access Manager Policy Server Host window opens, select **Access Manager Policy Server is installed on another machine**, enter the policy server host `phoenix` and port `7135`, and then click **Next**.

   b. When the Registry window opens, select LDAP and click **Next**.

   c. When the Domain Information window opens, we accepted `Default` (default-supplied value; do not change) in the Local domain name field and then clicked **Next**.

   d. When the LDAP Server Information window opens, we entered the host name and port and then clicked **Next**.

   e. When the SSL with the registry server window opens, we selected the appropriate setting regarding SSL usage and clicked **Next**.

   f. When the Logging Information window opens, we selected **Enable Tivoli Common Directory for logging**, entered log directory `c:\ibm\Tivoli\common`, and then clicked **Next**.

   g. When the Configuration Review window opens, note the policy server certificate path c:\ibm\Tivoli\TAM\keytab\pdcacert_download.b64, and click **Finish**.

3. From the Access Manager Configuration utility, select the Access Manager WebSEAL package and click **Configure**:

   a. When the Access Manager WebSEAL Configuration window opens, click **Configure**.

   b. When the Instance Identification window opens, enter `default` and then click **Next**. This option has a limit of 20 characters.

   c. When the WebSEAL Server Information page opens, enter the host name and port `7234` and then click **Next**.

   > **Note:** When the instance is created, the *server name* will be generated as <instance_name>-webseald-<hostname> (for example, default-webseald-bombay).

d. When the Administrator Identification window opens, enter the administrator `sec_master` and its password and then click **Next**.

e. When the SSL communications with LDAP Server window opens, select appropriate the SSL setting and click **Next**.

f. When the HTTP/HTTPS/Document Root Properties window opens, we accepted the following defaults and clicked **Finish**:

  - Allow HTTP access with port 80.
  - Allow HTTPS access with port 443.
  - Document root directory: `c:/ibm/Tivoli/PDWeb/www-default/docs` Where default is the instance name.

g. You should see the default instance created. When done, click **Close**.

4. Configure the Tivoli Access Manager Java Runtime Environment:

a. Select the Access Manage Java Runtime Environment from the Access Manager Configuration window. Click **Configure**.

b. When the Configuration Type window opens, select **Standalone** (only used by the local iKeyman utility) and click **Next**.

c. Enter the JRE path (for example, c:\ibm\Java131\jre), and then click **Next**.

d. When the Logging Information window opens, we selected **Enable Tivoli Common Directory for logging**, entered log directory `c:\ibm\Tivoli\common`, and then clicked **Finish**.

e. After configuration, click **OK**.

f. When done, close the configuration utility by clicking **Close**.

5. Configure GSKit iKeyman utility:

a. Navigate to the c:\ibm\Java131\jre\lib\security directory on the reverse proxy node.

b. Back up the java.security file to java.security.org.

c. Modify the java.security file as follows to add the IBM JCE and IBM CMS security providers:

```
security provider.3=com.ibm.crypto.provider.IBMJCE
security provider.4=com.ibm.spi.IBMCMSProvider
```

d. Save and close the java.security file.

e. Verify that the iKeyman utility starts properly from a command window by entering the following commands:

```
cd \ibm\gsk7\bin
set JAVA_HOME=c:\ibm\Java131\jre
gsk7ikm
```

f. Close the IBM iKeyman utility.

6.3.4 Installing Tivoli Access Manager V5.1 Base Fix Pack 2

In our example, we installed the Tivoli Access Manager Java Runtime Environment on the reverse proxy node. As part of our configuration, we need to upgrade the Tivoli Access Manager V5.1 Base to the Fix Pack 2 (5.1.0.2) level.

Ensure that the Tivoli Access Manager WebSEAL Windows service is stopped prior to installing the fix pack. For details about installing Tivoli Access Manager V5.1 Base Fix Pack 2, refer to 6.2.4, "Installing Tivoli Access Manager V5.1 Base Fix Pack 2" on page 175.

6.3.5 Installing Tivoli Access Manager V5.1 WebSEAL Fix Pack 2

To install Tivoli Access Manager V5.1 WebSEAL Fix Pack 2, complete the following steps.

> **Note:** For more detailed information about installing the Tivoli Access Manager V5.1 WebSEAL Fix Pack 2, refer to the *readme*, available at:
>
> `http://www.ibm.com/support/docview.wss?rs=203&context=SW000&q1=%2bfix+%2bTiv`
> `oli+%2bAccess+%2bManager&uid=swg24006926&loc=en_US&cs=utf-8&cc=us&lang=all`

1. Ensure that you are logged in to the reverse proxy node as an administrator.

2. Back up the Tivoli Access Manager V5.1 WebSEAL databases on the reverse proxy node be before installing Fix Pack 2 as follows:

   a. Create a database backup directory (for example, c:\ibm\websealdb.bak).

   b. Open a command window and enter the following command:

   ```
   pdbackup -action backup -list c:\ibm\Tivoli\tam\etc\pdbackup.lst -path
   c:\ibm\websealdb.bak
   ```

3. Ensure that you have installed IBM GSKit V7.0.1.16, which is a prerequisite to Fix Pack 2.

4. Ensure that the Tivoli Access Manager WebSEAL-<instance> Windows service is stopped before installing the fix pack.

5. Download Tivoli Access Manager V5.1 WebSEAL Fix Pack 2 from the following URL to a temporary directory (for example, c:\temp\tam51ws.fp2) on the reverse proxy node:

   `http://www.ibm.com/support/docview.wss?rs=203&context=SW000&q1=%2bfix+%2bTi`
   `voli+%2bAccess+%2bManager&uid=swg24006926&loc=en_US&cs=utf-8&cc=us&lang=all`

6. Navigate to the temporary directory where you downloaded the fix pack (for example, c:\temp\tam51ws.fp2) and run **5.1-AWS-FP02-WIN.exe** to start the Access Manager V5.1 WebSEAL Fix pack 2 Setup (5.1.0.2).

7. When the Welcome window opens, click **Next**.

8. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

9. When the installation completes, click **Finish**.

> **Note:** If a configuration window opens for the PDJRTE, you can click **Cancel**, because we already configured the Tivoli Access Manager Java Runtime Environment.

10. To verify that the fix pack installation was successful, open a command window and enter the command `pdversion`. This should return a list of components installed at the 5.1.0.2 level (Tivoli Access Manager Base and WebSEAL Fix Pack 2).

11. Start the Access Manager Auto-Start Service Windows service.

This completes the base installation and configuration of the reverse proxy node.

6.4 Java Runtime Environment on WebSphere Portal

To configure WebSphere Portal to work with IBM Tivoli Access Manager, you must install the IBM Tivoli Access Manager Java Runtime Environment.

The Java Runtime Environment included with IBM WebSphere Application Server V5.0.2.3 (Base) is incompatible with the `SvrSslCfg` command included with the Tivoli Access Manager Java Runtime Environment (PDJRTE). For this reason, we installed the Java Runtime Environment V1.3.1. For details, refer to "Installing Java Runtime Environment (JRE) V1.3.1" on page 225.

To install and configure the Tivoli Access Manager Java Runtime Environment (PDJRTE) using the native installation utility, complete the following steps on the WebSphere Portal server node:

1. Ensure that the following Windows services are started on the policy server node:

   – Tivoli Directory Server V5.2
   – Tivoli Access Manager policy server

2. Insert the *Tivoli Access Manager Base for Windows NT, Windows XP, Windows 2000 and Windows 2003* CD.

3. Navigate to the <CD_Root>\windows\PolicyDirector\Disk Images\Disk1 folder and run `Setup.exe` to start the installation.

4. When the Choose Setup Language window opens, select the language for the installation process and click **OK**.

5. When the Welcome window opens, click **Next**.

6. When the License Agreement window opens, click **Yes** to continue.

7. When the Select Packages window opens, we only selected the Access Manager Java Runtime Environment (PDJRTE) package and then clicked **Next**.

8. Choose the destination folder; we used C:\ibm\Tivoli\tam. Click **Next**.

9. Review your settings and click **Next**.

10. When the installation is complete, you should see a dialog box with the message "Installation completed successfully." Click **OK** to exit.

11. Install Tivoli Access Manager V5.1 Base Fix Pack 2 from the downloaded copy that you have from 6.2.4, "Installing Tivoli Access Manager V5.1 Base Fix Pack 2" on page 175.

12. Navigate to the temporary directory where you downloaded the fix pack and run **5.1-TAM-FP02-WIN.exe** to start the Access Manager V5.1 Fix Pack Setup (5.1.0.2).

13. When the Welcome window opens, click **Next**.

14. When the License Agreement window opens, review the terms and, if in agreement, click **Yes**.

15. When the installation is complete, click **Finish**.

> **Note:** This configuration is needed so that the WebSphere Application Server can use the Tivoli Access Manager APIs.

16. Enter the following command from c:\ibm\Tivoli\tam\sbin, in our example:

```
pdjrtecfg -action config -config_type full -host
phoenix.itsc.austin.ibm.com -port 7135 -java_home
c:\WebSphere\AppServer\java\jre
```

You should see the following message:

```
Configuration of Access Manager Java Runtime Environment completed
successfully.
```

17. Enter the following command from c:\ibm\Tivoli\tam\sbin, in our example:

```
pdjrtecfg -action config -config_type full -host
phoenix.itsc.austin.ibm.com -port 7135 -java_home c:\ibm\Java131\jre
```

You should see the following message:

```
Configuration of Access Manager Java Runtime Environment completed
successfully.
```

6.5 Enabling SSL between WebSEAL and WebSphere Portal

The our sample uses an external IBM HTTP Server for WebSphere Portal on the WebSphere Portal server node. This section describes how to configure mutual SSL between the WebSEAL on the reverse proxy node and the IBM HTTP Server on the WebSphere 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. For these reasons and limitations, when creating a WebSEAL junction, we needed to enable mutual SSL.

The section is organized into the following tasks:

► Enabling SSL for the WebSphere Portal server machine

► Importing IBM HTTP Server certificate into WebSEAL keystore

► Exporting the WebSEAL certificate

► Importing WebSEAL certificate into IBM HTTP Server keystore

► Enabling mutual SSL for IBM HTTP Server

6.5.1 Enabling SSL for the WebSphere Portal server machine

For the SSL configuration in the WebSphere Portal server machine, you can perform the configuration for the following components:

► The IBM HTTP Server SSL configuration is similar to the steps provided in 5.4.1, "Configuring IBM HTTP Server" on page 146. You can use the certificate signed by the Domino CA, the HTTPsigned.arm file.

► WebSphere Application Server SSL configuration is discussed in 5.4.2, "Configuring WebSphere Application Server" on page 148.

► WebSphere Portal server SSL configuration is discussed in 5.4.3, "Configuring SSL in WebSphere Portal" on page 149.

6.5.2 Importing IBM HTTP Server certificate into WebSEAL keystore

In our example, the 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 CMS security providers. As part of our configuration of the GSKit iKeyman utility, we added the IBM JCE and IBM CMS security providers. See 6.3.3, "Tivoli Access Manager: Configuring WebSEAL" on page 180.

To import the IBM HTTP Server certificate into the WebSEAL keystore on the reverse proxy node, complete the following steps:

1. Determine the WebSEAL keystore file name and location:

   a. Navigate to the C:\ibm\Tivoli\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 (ours is located in C:/ibm/Tivoli/PDWeb/www-default/certs/pdsrv.kdb).

2. Copy the IBM HTTP Server certificate, HTTPsigned.arm, from the WebSphere Portal server node to the C:/ibm/Tivoli/PDWeb/www-default/certs directory on the reverse proxy node.

3. Start the iKeyman utility on the reverse proxy node by entering the following commands from a Windows command window:

```
cd \ibm\gsk7\bin
set JAVA_HOME=c:\ibm\Java131\jre
gsk7ikm
```

4. From the menu bar, select **Key Database File → Open** and open the WebSEAL key file in C:/ibm/Tivoli/PDWeb/www-default/certs/pdsrv.kdb as a CMS key file. The default password for the key file is pdsrv.

5. From the Key database content drop-down list, select **Signer Certificates** and click **Add**.

6. When the Add CA's Certificate from a file window opens, we chose the following and then clicked **OK**:

   – Data type: **Base64-encoded ASCII data**
   – Certificate file name: `wp_httpd_cert.arm`
   – Location: `c:/ibm/Tivoli/PDWeb/www-default/certs`

7. When prompted for the label, we entered `WP HTTP Server SSL Key` and then clicked **OK**. For consistency, we entered the same name as used when we created the key.

   You should see the newly imported certificate listed (for example, WP HTTP Server SSL Key) among the Signer Certificates.

6.5.3 Exporting the WebSEAL certificate

In this section, we describe how to create a self-signed certificate for WebSEAL and how to export the certificate. Although WebSEAL does provide 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 V5.1 available to all customers and is thus not secure. For this reason, we create a new self-signed certificate.

To create a self-signed certificate for WebSEAL, complete the following steps:

1. If you closed the iKeyman utility after the previous step, you need to start the iKeyman utility and open the key database file (pdsrv.kdb).

2. Select **Personal Certificates** from the Key database content drop-down list.

3. From the menu bar, select **Create → New Self Signed Certificate**.

4. When the Create New Self Signed Certificate window opens, we used the following values and then clicked **OK**:

   – Key Label: `WebSEAL default key`
     In this case, default is the name of the WebSEAL instance.
   – Common Name: `bombay.itsc.austin.ibm.com`
   – Organization: `IBM`

5. When prompted, 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.

To export the WebSEAL certificate, complete the following steps:

1. If you closed the iKeyman utility after the previous step, you need to start the iKeyman utility and open the key database file (pdsrv.kdb).

2. Under Personal Certificates, select the certificate you just created in the previous steps (for example, WebSEAL default key).

3. Click **Extract Certificate**.

4. When the Extract Certificate to a File window opens, we chose the following and then clicked **OK**:

   – Data Type: **Base64-encoded ASCII data** (default)
   – Certificate file name: `webseal_default_cert.arm`
   – Location: `c:/ibm/Tivoli/PDWeb/www-default/certs`

5. Close the iKeyman utility.

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 file to define the new key label:

1. Navigate to the c:/ibm/Tivoli/PDWeb/etc directory.

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 (Access Manager WebSEAL default Windows service).

6.5.4 Importing WebSEAL certificate into IBM HTTP Server keystore

To import the WebSEAL certificate into the IBM HTTP Server keystore, complete the following steps:

1. Copy the WebSEAL certificate (for example, webseal_default_cert.arm) from the reverse proxy node to the c:/ibm/IBMHttpServer/ssl directory on the WebSphere Portal server node.

2. Start the IBM iKeyman utility on the WebSphere Portal server node by selecting **Start → Programs → IBM HTTP Server 1.3.26 → Start Key Management Utility**.

3. From the menu bar, select **Key Database File → Open**.

4. Select **c:\ibm\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**.

7. Click **Add**.

8. When the Add CA's Certificate from a file window opens, we chose the following and then clicked **OK**:

   – Data type: **Base64-encoded ASCII data**
   – Certificate file name: `webseal_default_cert.arm`
   – Location: `c:\ibm\IBMHttpServer\ssl`

9. When prompted for the label, we entered `WebSEAL default key` and then clicked **OK**. For consistency, we entered the same name as used when we created the key.

   You should see the newly imported certificate listed (for example, WebSEAL default key) among the **Signer Certificates**.

10.Close the iKeyman utility.

6.5.5 Enabling mutual SSL for IBM HTTP Server

To enable mutual SSL for IBM HTTP Server on the WebSphere Portal server node, complete the following steps:

1. Stop the IBM HTTP Server V1.3.26 Windows service.

2. Open the c:\ibm\IBMHTTPServer\conf\httpd.conf file with a text editor.

3. Search for the keyword `SSLClientAuth` inside the <VirtualHost entry.

4. Add the following entry after the commented SSLClientAuth entries:

   `SSLClientAuth required`

5. Save the changes to the httpd.conf file.

6. Restart the IBM HTTP Server.

> **Note:** After the SSLClientAuth required keyword and value are set, we will no longer be able to directly connect to the IBM HTTP Server through HTTPS. HTTPS connections to the IBM HTTP Server will only be permitted from WebSEAL.

7. Verify that you are not able to access the IBM HTTP Server directly through HTTPS. We did this by entering the following URL in a Web browser:

   `https://pretoria.itsc.austin.ibm.com`

8. You should see a message similar to the one shown in Figure 6-3. Click **Cancel**. If you are then prompted with a Security Alert, click **No**.
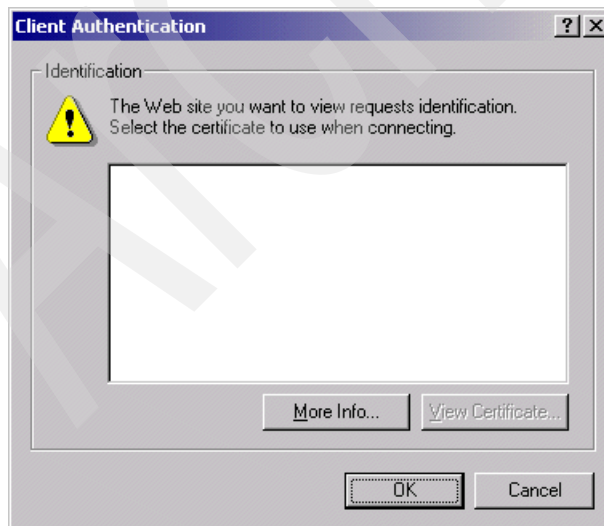
Figure 6-3   Client Authentication message

> **Note:** When we create a WebSEAL junction in 6.7.3, "Creating a WebSEAL junction" on page 198, we enable mutual SSL for the WebSEAL junction.

6.6 Configuring WebSphere Portal for access authorization

This section describes how to provide access authorization using Tivoli Access Manager for WebSphere Portal. The section is organized as follows:

► Configuring SSL between WebSphere Portal and Tivoli Access Manager

► Implementing JAAS authentication

► Modifying WebSphere Portal configuration files

► Verifying entries in Tivoli Access Manager for WebSphere Portal external authorization

6.6.1 Configuring SSL between WebSphere Portal and Tivoli Access Manager

The `SrvSslCfg` command is used to configure the SSL connection between WebSphere Application Server and Tivoli Access Manager. This command creates a keyfile and a properties file, which will be used later for the WebSphere Portal configuration.

> **Note:** We found that the Java Runtime Environment included with IBM WebSphere Application Server V5.0.2.3 (Base) is incompatible with the `SvrSslCfg` command. For this reason, we installed the Tivoli Access Manager Java Runtime Environment V1.3.1 and configured the Tivoli Access Manager Runtime to use the stand-alone IBM JRE V1.3.1.

The `SrvSslCfg` command also creates the WebSphere administrative user and inserts this user in the following Tivoli Access Manager LDAP groups:

► cn=remote-acl-users
► cn=SecurityGroup,secauthority=Default

To configure the SSL connection between WebSphere Application Server used by WebSphere Portal and the Tivoli Access Manager, issue the command shown in Example 6-1 on page 191.

Example 6-1 shows the complete command that we issued. Due to the length of the command, we edited the command into a batch file so that we can easily verify it and also reuse the command.

Example 6-1   Running the SvrSslCfg command

```
c:\ibm\Java131\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 phoenix.itsc.austin.ibm.com:7135:1
-authzsvr phoenix.itsc.austin.ibm.com:7136:1
-cfg_file "c:\WebSphere\AppServer\java\jre\PdPerm.properties"
-key_file "c:\WebSphere\AppServer\java\jre\lib\security\pdperm.ks"
-cfg_action replace
```

The following list describes the arguments for the command:

-action	This defines the action that we want to perform. Here, we configure the SSL service.
-admin_id	The Tivoli Access Manager administrator user, sec_master.
-admin_pwd	The password for the user ID in admin_id.
-appsvr_id	The unique WebSphere Application Server user ID to be created in Tivoli Access Manager. We use amwas.
-port	The port number on which WebSphere Application Server listens for policy server notifications. This value must be filled in even though WebSphere Application Server does not currently use this port.
-mode	The configuration mode, whether local or remote authentication.
-policysvr	The host and port for the policy server for the Tivoli Access Manager pdmgrd process.
-authzsvr	The host and port for the authorization server for the Tivoli Access Manager pdacld process.
-cfg_file	The configuration file that will be created and inserted into ExternalAccess ControlService.properties.
-key_file	The path to the keystore file.
-cfg_action	This indicates the action option. We use replace for replacing any existing configuration and keystore file.

When the **SvrSslcfg** command completes, you should see the following message:

```
The configuration completed successfully.
```

To verify that the **SvrSslCfg** command worked properly, run the **pdadmin** command from the policy server node. You should see the newly created server process, in our case, amwas-pretoria, from the output of the **server list** command.

6.6.2 Implementing JAAS authentication

In this section, we configure WebSphere Portal to extract and cache the Tivoli Access Manager Java Authentication and Authorization Service (JAAS) credential from the HTTP header data sent by WebSEAL. This step 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. To implement JAAS authentication, complete the following steps:

1. Modify the WebSphere Portal server configuration files to enable JAAS as follows:

   – Modify the ConfigService.properties file from the WebSphere Portal server node in the c:\WebSphere\PortalServer\shared\app\config\services directory and add the following line in the file:

   ```
   execute.portal.jaas.login=true
   ```

   – Modify the callbackheaderslist.properties file from the WebSphere Portal server node in the c:\WebSphere\PortalServer\shared\app\config directory and uncomment the following entries:

   ```
   header.1=iv-user
   header.2=iv-creds
   ```

   – Modify the ExternalAccessControlService.properties file from the WebSphere Portal server node in the c:\WebSphere\PortalServer\shared\app\config\services directory and add the following line in the file:

   ```
   externalaccesscontrol.pdurl=file:///c:/WebSphere/AppServer/java/jre/PdPe
   rm.properties
   ```

2. Configure JAAS in WebSphere Application Server.

   JAAS can be configured within WebSphere Application Server on the WebSphere Portal server node by using the WebSphere Administrative Console or by using a WebSphere JACL command script. A sample JACL script is provided in config.was.jaas.jacl. The configuration adds the 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.

This procedure uses the WebSphere Application Server Administrative Console. The objective is to define Tivoli Access Manager modules for JAAS authorization. It is performed under the path **Security** → **JAAS Configuration** → **Application Logins**. The modules and additional parameters are shown in Table 6-1.

Table 6-1   Tivoli Access Manager authorization module settings

Module type	Module class name	Authentication strategy	Custom property: delegate
Portal_Login JAAS Login Modules	com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy	Required	com.ibm.wps.sso.WebSealLoginModule
Portal_Login JAAS Login Modules	com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy	Required	com.tivoli.mts.PDLoginModule
Portal_SubjectRebuild JAAS Login Modules	com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy	Required	com.ibm.wps.sso.WebSealLoginModule
Portal_SubjectRebuild JAAS Login Modules	com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy	Required	com.tivoli.mts.PDLoginModule

6.6.3 Modifying WebSphere Portal configuration files

This section includes the following modifications to the WebSphere Portal configuration files to enable external authorization using Tivoli Access Manager:

1. Modify the ExternalAccessControlService.properties file from the <wp_home>/shared/app/config/services directory. Example 6-2 includes our sample values.

Example 6-2   ExternalAccessControlService.properties example

```
externalaccesscontrol.ready=true
externalaccesscontrol.server=WebSphere_Portal
externalaccesscontrol.application=WPS
externalaccesscontrol.cell=cell
externalaccesscontrol.pdroot=/WPSv5
externalaccesscontrol.pduser=sec_master
externalaccesscontrol.pdpw=<password>
externalaccesscontrol.pdurl=file:///c:/IBM/WebSphere/AppServer/java/jre/PdPerm.
properties
externalaccesscontrol.createAcl=true
externalaccesscontrol.pdactiongroup=[WPS]
```

Use the WebSphere Application Server encoding mechanism **PropFilePasswordEncoder.bat** to mask the password that will appear in the ExternalAccessControlService.properties file. For example:

```
c:\WebSphere\AppServer\bin\PropFilePasswordEncoder.bat
c:\WebSphere\PortalServer\shared\app\config\services\ExternalAccessControlS
ervice.properties externalaccesscontrol.pdpw
```

This command generates a backup file (bak file) with the original clear text password. You might want to remove this file.

2. Modify the AccessControlConfigService.properties file from the <wp_home>/shared/app/config/services directory. Ensure that you have the following line:

```
accessControlConfig.enableExternalization=true
```

3. Modify the services.properties file from the <wp_home>/shared/app/config directory. Only the last entry that is shown in bold needs to be modified, as shown in Example 6-3.

Example 6-3   services.properties snippet example

```
com.ibm.wps.ac.impl.AccessControlDataManagementService=com.ibm.wps.ac.impl.Acce
ssControlDataManagementServiceImpl
com.ibm.wps.services.ac.PermissionFactoryService=com.ibm.wps.ac.impl.Permission
FactoryImpl
com.ibm.wps.services.ac.ACPrincipalFactoryService=com.ibm.wps.ac.impl.ACPrincip
alFactoryImpl
com.ibm.wps.services.ac.internal.AccessControlConfigService=com.ibm.wps.ac.impl
.AccessControlConfigImpl
com.ibm.wps.services.ac.AccessControlService=com.ibm.wps.ac.impl.AccessControlI
mpl
com.ibm.wps.services.ac.ExternalAccessControlService=com.ibm.wps.ac.esm.TAMExte
rnalAccessControlImpl
```

4. Modify the AccessControlDataManagementService.properties file from the <wp_home>/shared/app/config/services directory. Add or modify the following values:

```
accessControlDataManagement.enableNestedGroups=false
accessControlDataManagement.cacheTimeout=30
accessControlDataManagement.externalizeAllRoles=false
accessControlDataManagement.createAdminMappingXMLAccess=true
```

Where:

enableNestedGroups   Tivoli Access Manager V5.1.0.2 does not support nested groups, while WebSphere Portal V5.0.2 does. We disable this to adhere to the Tivoli Access Manager limitation.

cacheTimeout	Portal Access Control maintains caches for better performance of requests. This property automatically invalidates the Portal Access Control caches after the given time (in seconds).
externalizeAllRoles	This property is only applicable for externalization of resources through the user interface. Setting this value to false allows selective external creation of resources and roles.
createAdminMappingXMLAccess	
	This property is only applicable for externalization of resources through XMLAccess. Setting this value to true externalizes all the admin roles through XML access.

5. Reorder the role names by resource type (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, edit the AccessControlDataManagementService.properties file from the <wp_home>/shared/app/config/services directory. Modify the file as follows:

   `accessControlDataManagement.reorderRoleNames=true`

6. Restart the WebSphere_Portal application server (you can defer this step if you are planning to implement the reorderRoleNames property).

6.6.4 Verifying entries in Tivoli Access Manager for WebSphere Portal external authorization

When WebSphere Portal starts, TAMExternalAccessControlServices creates the necessary topology in IBM Tivoli Access Manager to begin externalizing roles and also creates the core WPS_Administrator-Virtual_wps-EXTERNAL ACCESS CONTROL_1 role. It also creates an access control list (ACL) and 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 issue the `pdadmin -a sec_master -p <password>` command.

2. Verify the that the /WPSv5 objectspace has been created by entering the `objectspace list` command. You should see the/WPSv5 objectspace in the list.

3. Verify that the WPS action group has been created by entering the `action group list` command. 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 representing 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]
   ```

6.7 Configuring WebSphere Portal authentication

In our example, our runtime is currently configured for users to directly log in to WebSphere Portal on the WebSphere Portal server node. After completing this section, the authentication for WebSphere Portal will be performed by a combination of WebSEAL on the reverse proxy node and the Tivoli Access Manager policy server on the policy server node.

This section includes the following tasks:

► Applying Tivoli Access Manager ACLs to new LDAP suffixes

► Defining additional MIME types for WebSphere Application Server

► Creating a WebSEAL junction

► Enabling forms authentication on WebSEAL

► Importing WebSphere Portal users and groups into Tivoli Access Manager

► Defining access controls for WebSphere Portal URIs

► Configuring the junction mapping table

► Configuring SSO for WebSEAL and WebSphere through TAI

► Activating the LTPA junction with WebSEAL

► Configuring WebSphere Portal login and logout for WebSEAL

6.7.1 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 `o=ibm,c=us` after we configured Tivoli Access Manager. For this reason, we must manually apply Tivoli Access Manager ACLs to the suffix.

> **Note:** For more information about applying Tivoli Access Manager ACLs to a new LDAP suffix, refer to Appendix D, "Managing user registries," in *IBM Tivoli Access Manager Base Administration Guide, V5.1*, SC32-1360.

Here, we describe how to apply Tivoli Access Manager ACLs to a new LDAP suffix using an LDIF file import in the Tivoli Directory Server. Our sample code includes c:\6325code\config\ldap\tam-acls.ldif. If you choose to import ACLs through the ldif, you can skip the rest of this section.

To apply IBM Tivoli Access Manager ACLs to a new LDAP suffix, complete the following steps:

1. Copy the c:\6325code\config\ldap\tam-acls.ldif file to the c:\temp directory.

2. Modify the tam-acls.ldif file for your suffix.

3. From the command line, execute the following command. In our example:

```
ldapmodify -h phoenix.itsc.austin.ibm.com -D cn=root -w <password> -i
c:\temp\tam-acls.ldif
```

6.7.2 Defining additional MIME types for WebSphere Application Server

By default, WebSphere Application Server V5 is not configured with MIME types for Java Archive 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 6-2 on page 198 lists the MIME type definitions that we add in this section. If your portlet application uses other MIME types not found by default within WebSphere Application Server, follow the same procedure to add the MIME type definitions.

Table 6-2   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, complete the following steps:

1. Ensure that the server1 application server is started on the WebSphere Portal server node.

2. Open the WebSphere Application Server Administrative Console:

   a. In our example, we entered the URL:
      `https://pretoria.itsc.austin.ibm.com:9043/admin`

   b. Enter the WebSphere administrator user ID and password (for example, wpsbind).

3. Select **Environment** → **Virtual Hosts** and select **default_host**.

4. Add the new MIME type. Click **MIME Types** under Additional Properties and click **New**.

5. In the New MIME Type window, enter the information in Table 6-2, and then click **OK**.

6. Click **Save**. In the Save to Master Configuration window, click **Save**.

6.7.3 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 our example, we chose to create an input file. In our example, we use the multiple command mode and type the commands to create the junction.

Creating the junction for WebSphere Portal

To create a junction for WebSphere Portal, complete the following steps:

1. Ensure that the Access Manager policy server Windows service is started.

2. Start the **pdadmin** command line interface on the reverse proxy node by selecting **Programs → IBM Tivoli Access Manager → Administration Command Prompt**.

3. Log in by entering `login` at the pdadmin prompt. When prompted, enter the user ID `sec_master` and `<password>`.

4. To get a list of servers, enter the command server list, and it should return a list of servers with the WebSEAL host such as:

   ```
   default-webseald-bombay
   ```

5. Run the server task command that will enable mutual SSL for the WebSEAL junction created. For our example, this is:

   ```
   server task default-webseald-bombay create -t ssl
           -h pretoria.itsc.austin.ibm.com -p 443 -j -w -c all
           -K "WebSEAL default key"
           -D "CN=pretoria.itsc.austin.ibm.com,O=IBM,C=US" /portal
   ```

   Where the parameters are as follows:

   default-webseald-bombay

   The WebSEAL server name as retrieved using the **server list** command in step 4.

   -t ssl   The junction type, which is either TCP or SSL.

   -h <hostname>   Hostname is the back-end server host name. This is our WebSphere Portal server node, pretoria.

   -p 443   The SSL port is the port that the back-end server used. This is the HTTP server port, as opposed to the WebSphere SSL port of 9444.

   -j   This option enables junction cookies for handling server-relative URLs. This also applies to the mapping of the dynamically generated URLs, such as from applet or ActiveX control.

   -w   This option establishes support for Windows file systems.

   -c all   This option can be **-c all** or **-c iv-user** for WebSphere Portal integration. When only performing authentication, **-c -iv-user** is sufficient. When using performing authentication and authorization, **-c all** must be used.

   -K <label>   This is the WebSEAL key label on the reverse proxy node.

-D \<stanza\>	This is the LDAP distinguished name of the server. Our example is: `CN=pretoria.itsc.austin.ibm.com,O=IBM,C=US`.
/portal	This is the portal's junction name.

> **Note:** When creating the junction, WebSEAL will attempt to connect to the back-end system. If the system is not available, you will see the following messages:
>
> ```
> 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 will still be created.

6. Verify that the junction was created properly. Using the `pdadmin` utility, enter the `server task default-webseald-bombay list` command and make sure that /portal is listed as a junction. Furthermore, you can see all the settings for the /portal junction using the `server task default-webseald-bombay show /portal` command.

6.7.4 Enabling forms authentication on WebSEAL

By default, WebSEAL uses HTTP basic authentication for its authentication challenge to the user. With basic authentication, the logout action 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 will need to update the webseald-default.conf file (the default is the WebSEAL instance name) on the reverse proxy node as follows:

1. Back up the webseald-default.conf file from the C:\ibm\Tivoli\PDWeb\etc directory.

2. We modified the webseald-default.conf file, as shown in the following examples, for our example.

Example 6-4   Form authentication settings in webseald-default.conf

```
[ba]
ba-auth = none

[forms]
forms-auth = https
```

Example 6-5   Our example URL filtering settings in webseald-default.conf

```
[script-filtering]
script-filter=yes
```

Example 6-6   Additional webseald-conf file configurations

```
[server]
dynurl-allow-large-posts=yes

[junction]
http-timeout=300
https-timeout=300

[session]
ssl-id-sessions=no
```

3. Restart the Access Manager WebSEAL-default Windows service to enable these configuration changes.

4. Save the webseald-default.conf file.

5. To verify that the forms authentication is enabled, we access the WebSEAL with a Web browser:

   `https://<webseal_hostname>`

   You should now get the WebSEAL login form page in the browser instead of the WebSEAL basic authentication pop-up dialog box.

6. Log on to the WebSEAL with the sec_master user ID and password. You should see the WebSEAL splash window.

Now that forms are enabled, users will be able to log out without needing to close the Web browser.

6.7.5 Importing WebSphere Portal users and groups into Tivoli Access Manager

Although the users and groups used by WebSphere Portal have already been created in the Tivoli Directory Server directory, the users have not been imported

into Tivoli Access Manager. The import of users into Tivoli Access Manager includes adding attributes to existing users in the LDAP directory.

In our example, we create a command file called wp-tam-user-import.pd, as shown in Example 6-7.

Example 6-7   Our example wp-tam-user-import.pd file

```
user import -gsouser wpsadmin uid=wpsadmin,cn=users,o=ibm,c=us
user modify wpsadmin account-valid yes
user modify wpsadmin password-valid yes
group import wpsadmins cn=wpsadmins,cn=groups,o=ibm,c=us
```

To import the WebSphere Portal users and groups into Tivoli Access Manager using a command file through **pdadmin**, enter the following commands:

```
pdadmin -a sec_master -p <password> wp-tam-user-import.pd
```

To verify that the users and groups where imported properly, enter the following commands from a Windows command window:

```
pdadmin -a sec_master -p <password> user list * 100
pdadmin -a sec_master -p <password> group list * 100
```

6.7.6 Defining access controls for WebSphere Portal URIs

This section describes how to define four access categories for WebSphere Portal, as defined in Table 6-3.

Table 6-3   Access categories for WebSphere Portal

Access category	Description
WP_all_access	Access for all users (authenticated and unathenticated)
WP_authenticated_access	Access for authenticated users only
WP_admin_access	Access for administrator users only
WP_no_access	No access

Creating Tivoli Access Manager objects for WebSphere Portal URIs

To create Tivoli Access Manager objects for the WebSphere Portal URIs, complete 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 *IBM Tivoli Access Manager WebSEAL Administration Guide V5.1*, SC32-1359.

1. Create a file named dynurl.conf in the c:\ibm\Tivoli\PDWeb\www-*default*\lib directory with the text from Example 6-8, where `default` is the name of the WebSEAL instance.

Example 6-8   Our 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 added entries for both *before* the mapping and *after* the mapping versions of URLs that will be handled by the junction mapping table (JMT). This is necessary, because WebSEAL will perform two ACL checks, one on the URL before the JMT transformation and one after. Both must pass for access to be granted.
>
> Given that the real access control check is the second one, we added 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 will now always pass.

2. To activate the definitions, you can issue the **server task default-webseald-bombay dynurl update** command from **pdadmin** or restart the WebSEAL service.

Defining the access control policy for WebSphere Portal

This section describes how to define the access control policy for WebSphere Portal and includes the following operations:

► Create Tivoli Access Manager ACL templates corresponding to the access categories defined for WebSphere Portal.

► Update the ACLs for the imported users and groups.

► Attach the ACLs to protected objects for WebSphere Portal.

We define a command file called wp-tam-acl.pd, as shown in Example 6-9 on page 204.

> **Note:** The wp-tam-acl.pd command file used in our example includes three sections:
>
> ► The first two sections in Example 6-9 on page 204 do not need to be modified unless you need to create new access categories.
>
> ► The last section in Example 6-9 on page 204 includes settings that will change based on your environment. For example, you will need to update the following: The /WebSEAL/host-instance_name represents the beginning of the Web space for a particular WebSEAL server instance (for example, /WebSEAL/bombay-default). To retrieve your setting, enter the following command on the reverse proxy node:
>
> ```
> pdadmin -a sec_master -p <password> object list /WebSEAL
> ```

Example 6-9   Our 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/bombay-default/portal/wps/config WP_admin_access
```

```
acl attach /WebSEAL/bombay-default/portal/wps/myportal WP_authenticated_access
acl attach /WebSEAL/bombay-default/portal/wps/portal WP_all_access
acl attach /WebSEAL/bombay-default/portal/wps/doc WP_all_access
acl attach /WebSEAL/bombay-default/portal/wps WP_all_access
```

Execute the command file wp-tam-acl.pd by entering the following command from a Windows command window:

```
pdadmin -a sec_master -p <password> wp-tam-acl.pd
```

6.7.7 Configuring the junction mapping table

Several portlets, including the Resource Permissions portlet, and the productivity components editors use relative JavaScript within the portlet or component. These portlets and components will 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 junction mapping table (JMT) function in WebSEAL.

To enable the JMT function, define a text file called jmt.conf on the reverse proxy node:

1. Create the jmt.conf file in the c:/ibm/Tivoli/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.

   Our example jmt.conf file is shown in Example 6-10.

Example 6-10   Our example jmt.conf file

```
/portal /wps/*
```

3. You can reload the JMT in WebSEAL using the **pdadmin** subcommand **server task default-webseald-bombay jmt load**.

4. Verify that the JMT (jmt.conf) is working properly. For our example, we entered the following URL in a Web browser to access WebSphere Portal through WebSEAL (reverse proxy node host name):

```
http://bombay.itsc.austin.ibm.com/wps/portal
```

You should see the default WebSphere Portal page for unauthenticated users.

6.7.8 Configuring SSO for WebSEAL and WebSphere through TAI

When using a reverse proxy such as WebSEAL to authenticate users in the DMZ, it is desirable that WebSphere Application Server, as well as other back-end applications and services, trust the authentication that has been performed and the identity that is being presented by the reverse proxy. If this trust can be established, users then need only authenticate once to the reverse proxy in order to have access to all authorized services located beyond that proxy. This is commonly known as reverse proxy single sign-on (RPSS).

There are two ways to establish a trust relationship between WebSphere Application Server and WebSEAL:

▶ Trust Association Interceptor (TAI)

This section describes the TAI implementation procedure.

▶ Lightweight Third Party Authentication (LTPA) token

For details about implementing this using an LTPA token for the single sign-on configuration, see 6.7.9, "Activating the LTPA junction with WebSEAL" on page 209.

There are three possible methods of verifying 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
  This is the method used in our example.

Enabling TAI on the WebSphere Portal server node

To enable TAI in WebSphere Application Server on the WebSphere Portal server node using the WebSphere Application Server Administrative Console, complete the following steps.

> **Note:** As an alternative to enabling TAI using the WebSphere Application Server Administrative Console as described in this section, you can use a WebSphere Application Server admin script.

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

2. Start the WebSphere Application Server Administrative Console:

   a. For our example, we entered the URL:

   ```
   https://pretoria.itsc.austin.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. Click **Trust Association** under Additional Properties.
5. In the Trust Association window, select the **Trust Association Enabled** option. Click **Apply**.
6. Click **Interceptors** under Additional Properties.
7. Click **com.ibm.ws.security.web.WebSealTrustAssociationInterceptor**.
8. Click **Custom Properties** under Additional Properties.

Click **New** and enter the General Properties name and value pairs specified in Table 6-4. For more information about the possible values that these properties might have, refer to the *WebSphere Portal Information Center*, available at:

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

Table 6-4   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	bombay.itsc.austin.ibm.com, bombay **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 WebSphere Portal server node.

After you have created all these properties, the Custom Properties should look as shown in Figure 6-4 on page 208.
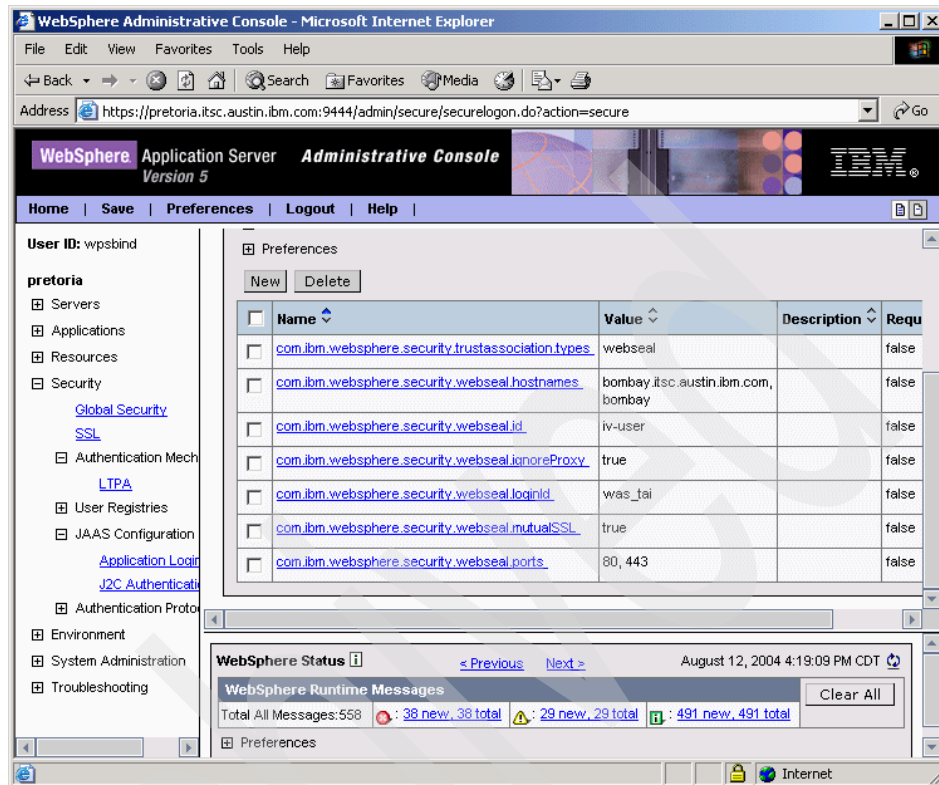
Figure 6-4   Trust Association properties

> **Tip:** Check and double-check the names *and* values of all the Custom
> Properties before saving the configuration changes.

9. Click **Save**. When the Save to Master Configuration window opens, click
   Save.

10. Click **Logout**.

11. Restart the WebSphere_Portal application server.

Verifying the TAI configuration

Now that we have enabled TAI within WebSphere Application Server on the WebSphere Portal server node, we recommend that you verify that TAI is working properly by completing the following steps:

1. Verify that access to the unauthenticated portal pages is working properly. For our example, we entered the following in a Web browser:

   ```
   http://bombay.itsc.austin.ibm.com/portal/wps/portal
   ```

   In this case, there should be no authentication.

2. Verify that authenticated access is working properly. For our example, we entered the following URL in a Web browser:

   ```
   https://bombay.itsc.austin.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.

> **Note:** The Logout link will not work at this stage. In the next section, we configure the WebSphere Portal login and logout for use with WebSEAL.
>
> If you are directed to the Portal login page at wps/portal/.scr/Login or the public page, there is a problem with the Trust Association Interceptor configuration.

6.7.9 Activating the LTPA junction with WebSEAL

To activate the LTPA junction, complete the following steps:

1. Copy the LTPA key to WebSEAL.

   Copy the LTPA key you created in 4.4.5, "Configuring single sign-on" on page 95 when you enabled single sign-on between WebSphere Portal and your Domino Extended Products to the www-default certs directory on your reverse proxy server (C:\ibm\Tivoli\PDWeb\www-default\certs in our example)

2. Create an LTPA WebSEAL junction.

   Run the `pdadmin` command line utility. For our example, we issued the following command:

   ```
   server task default-webseald-bombay create -t tcp -h
   pretoria.itsc.austin.ibm.com -p 80 -j -w -A -F
   "c:\ibm\Tivoli\PDWeb\www-default\certs\ltpa.key" -Z <ltpa_password> /portal
   ```

3. You can verify that the junction was created properly. We used the following commands:

– The new junction /portal should be listed in the output of the command:

```
server task default-webseald-bombay list
```

– The detailed property of the junction can be shown using the command:

```
server task default-webseald-bombay show /portal
```

6.7.10 Configuring WebSphere Portal login and logout for WebSEAL

In our example, we 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. In this section, we describe how to configure WebSphere Portal login and 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, we modify the web.xml file in the WebSphere Portal server node from the c:/WebSphere/AppServer/installedApps/pretoria/wps.ear/wps.war/WEB-INF/ directory. Modify the web.xml file contents as shown in Example 6-11.

Example 6-11   Our 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>
```

Creating wpslogout.html

When a WebSphere Portal user log outs of WebSEAL, we would like to display the public (unauthenticated) portal page at:

```
http://bombay.itsc.austin.com/portal/wps/portal
```

To achieve this behavior, we must create the wpslogout.html file to redirect the WebSEAL logout to the WebSphere Portal public page. Create the wpslogout.html file in the c:\ibm\Tivoli\PDWeb\www-default\lib\html\C\ directory, as shown in Example 6-12 on page 211. You will need to modify the HREF value to include your WebSEAL host name.

Example 6-12   Our wpslogout.html example

```
<!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://bombay.itsc.austin.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://bombay.itsc.austin.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) access through the
WebSEAL perform logout, the logout.html page will be displayed. Edit the
logout.html file from the reverse proxy node in the
C:\ibm\Tivoli\PDWeb\www-default\lib\html\C directory. See the sample in
Example 6-13.

Example 6-13   Our logout.html example

```
<!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, edit the ConfigService.properties file in the
C:\WebSphere\PortalServer\shared\app\config\services directory. Update the
redirect.logout entries as shown in Example 6-14. You will need to update the
redirect.logout and redirect.logout.ssl to true, and redirect.logout.url for your
environment.

Example 6-14   Excerpt of ConfigService.properties

```
# Logout redirect parameters
#
# Default: false, false, <none>
redirect.logout     = true
redirect.logout.ssl = true
redirect.logout.url =
https://bombay.itsc.austin.ibm.com/pkmslogout?filename=wpslogout.html
```

Modifying ToolBarInclude.jsp

This section modifies the ToolBarInclude.jsp file. ToolBarInclude.jsp allows end
users to self register, request a password, and edit their profile. With the
WebSEAL frontend, these capabilities cannot be performed using WebSphere
Portal. Furthermore, the address of the Log in link needs to point to the reverse
proxy node.

There is one occurrence of the ToolBarInclude.jsp file in the
wps.war/themes/html directory. This ToolBarInclude.jsp file is used as the default
if a theme does not have its own ToolBarInclude.jsp file. You must also update
the ToolBarInclude.jsp file in the /wps.ear/wps.war/themes/html/<theme_name>
directory for each theme.

The theme directory is located under
c:/WebSphere/AppServer/installedApps/pretoria/wps.ear/wps.war/themes/html.
A sample ToolBarInclude.jsp file is shown in Example 6-15 on page 215.

Example 6-15   Our example TollBarInclude.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>
```

Open and save the version of the Default.jsp file found in the root of the html directory and corresponding theme directory. This is necessary to make the application server recompile the JSPs to have the changes in the ToolBarInclude.jsp file take effect.

Modifying the WpsHostName property in wpconfig.properties

After the junction configuration, the WpsHostName property in wpconfig.properties should be set to the WebSEAL host name. This will allow proper URL filtering and protocol switching by WebSEAL. The wpconfig.properties file is located in the WebSphere\PortalServer\config directory.

To load the configuration changes, issue the `wpsconfig httpserver-config` command from the config directory and restart the WebSphere_Portal application server.

6.8 Protecting Domino Extended Products

If IBM Tivoli Access Manager is protecting WebSphere Portal with a TAI junction, you have two choices for protecting the Domino Extended Products. You can choose not to protect the Domino Extended Products, and let the LTPA token generated by WebSphere handle all single sign-ons, or you can create LTPA junctions to all back-end Domino Extended Products accessed through WebSphere Portal.

If you decide not to protect the Domino Extended Products, you need to complete only the steps in 6.8.1, "Configuring Tivoli Access Manager to not protect the Domino Extended Products" on page 216.

If you decide to protect the entire Collaborative Solution, you need to complete the steps in 6.8.2, "Protecting the Domino mail and application servers with an LTPA junction" on page 217, 6.8.3, "Protecting Lotus Team Workplace with an LTPA junction" on page 217, and 6.8.4, "Protecting Lotus Instant Messaging and Web Conferencing with an LTPA junction" on page 219.

6.8.1 Configuring Tivoli Access Manager to not protect the Domino Extended Products

If you do not want IBM Tivoli Access Manager to protect the Domino Extended Products, you must configure Tivoli Access Manager WebSEAL to allow back-end domain cookies to be passed to the browser. These LTPA cookies generated by WebSphere Portal will be used for single sign-on to all the back-end Domino Extended Products. From the c:\ibm\Tivoli\PDWeb\etc

directory, back up the webseald-default.conf file and modify it to set the allow-backend-domain-cookies to yes, which enables all domain cookies passed to the client to remain unmodified.

6.8.2 Protecting the Domino mail and application servers with an LTPA junction

To protect the Domino mail and application servers with LTPA junctions, you will need to create a junction to each Domino mail and application server. To create the junction to Domino, complete the following steps:

1. Copy the LTPA key to WebSEAL.

   Copy the LTPA key you created in 4.4.5, "Configuring single sign-on" on page 95 when you enabled single sign-on between WebSphere Portal and your Domino Extended Products servers to the www-default certs directory on your reverse proxy server (C:\ibm\Tivoli\PDWeb\www-default\certs in our example).

2. Create the LTPA junction using the `pdadmin` command line interface from the reverse proxy node. We used the following command:

   ```
   server task default-webseald-bombay create -t tcp -h
   kingston.itsc.austin.ibm.com -p 80 -i -j -A -F
   "c:\ibm\Tivoli\PDWeb\www-default\certs\ltpa.key" -Z <ltpa_password> /dom
   ```

6.8.3 Protecting Lotus Team Workplace with an LTPA junction

To protect the Lotus Team Workplace servers with LTPA junctions, complete the following steps:

1. Copy the LTPA key to WebSEAL.

   Copy the LTPA key you created in 4.4.5, "Configuring single sign-on" on page 95 when you enabled single sign-on between WebSphere Portal and your Domino Extended Products servers to the www-default certs directory on your reverse proxy server (C:\ibm\Tivoli\PDWeb\www-default\certs in our example).

2. Create the LTPA junction using the `pdadmin` command line interface from the reverse proxy node. We used the following command:

   ```
   server task default-webseald-bombay create -t tcp -h
   kingston.itsc.austin.ibm.com -p 80 -i -j -A -F
   "c:\ibm\Tivoli\PDWeb\www-default\certs\ltpa.key" -Z <ltpa_password> /qp
   ```

3. Disable page compression in WebSEAL or Team Workplace.

   After configuring Lotus Team Workplace to be protected by Tivoli Access Manager, the My Places link will continue to redirect users directly to the Team Workplace server and not through the Tivoli Access Manager junction as it should. Sametime meetings created in Team Workplace will also have the same problem. The reason this occurs is because Team Workplace 6.5.1 uses page compression when sending HTTP pages from Team Workplace to the browser. When page compression is used, Tivoli Access Manager cannot read the page sent to the browser and filter the URLs as it should.

   There are two ways to resolve this issue. You can disable page compression in Team Workplace, or you can set WebSEAL to force Team Workplace to not use page compression when working with Tivoli Access Manager.

   – To disable page compression in Team Workplace, edit the qpconfig.xml file and uncomment and set the page compression section into:

   ```
   <page_compression enabled="false">
   </page_compression>
   ```

   – To set WebSEAL to not allow page compression, on the reverse proxy node, edit the webseald-default.conf file from the c:\ibm\Tivoli\PDWeb\etc directory as shown in Example 6-16. The accept-encoding option prevents junctioned servers from returning compressed data to WebSEAL.

Example 6-16   webseald changes to prevent compressed data from Team Workplace

```
[filter-request-headers]
#
# HTTP headers to filter from the client request before sending to the
# back-end web server.  Note that this list is in addition to headers
# that WebSEAL will always filter, eg iv-user, iv-groups.
#
# Format is:
#   header = <header-name>
#
# The header name is case insensitive.
#
# The addition of "accept-encoding" to this list will prevent junctioned
# servers from returning compressed data to WebSEAL.  WebSEAL cannot
# filter compressed data.
header = accept-encoding
```

6.8.4 Protecting Lotus Instant Messaging and Web Conferencing with an LTPA junction

To protect the Lotus Instant Messaging and Web Conferencing servers with LTPA junctions, complete the following steps:

1. Ensure that Instant Messaging and Web Conferencing is configured for tunneling.

   For Instant Messaging and Web Conferencing to work with WebSEAL, you must set the Instant Messaging and Web Conferencing server to tunnel every connection over port 80. If you did not select this option when you installed the Instant Messaging and Web Conferencing server, refer to *Sametime: How to Enable HTTP Tunnelling Over Port 80*, Technote 1090222, for information about how to set up the server for tunneling after it has been installed, available at:

   http://www.ibm.com/support/docview.wss?rs=0&uid=swg21090222

2. Copy the LTPA key to WebSEAL.

   Copy the LTPA key you created in 4.4.5, "Configuring single sign-on" on page 95 when you enabled single sign-on between WebSphere Portal and your Domino Extended Products servers to the www-default certs directory on your reverse proxy server (C:\ibm\Tivoli\PDWeb\www-default\certs in our example).

3. Create the LTPA junction using the **pdadmin** command line interface from the reverse proxy node. We used the following command:

```
server task default-webseald-bombay create -t tcp -h
laredo.itsc.austin.ibm.com -p 80 -i -j -A -F
"c:\ibm\Tivoli\PDWeb\www-default\certs\ltpa.key" -Z <ltpa_password> /st
```

> **Note:** You cannot use the **-w** parameter for this setup. Some requests generated by Instant Messaging and Web Conferencing are not allowed through the junction if the **-w** parameter exists. You must also ensure that the LTPA key used in the junction is the same LTPA key that the Instant Messaging and Web Conferencing server uses in its Web SSO Configuration document.

4. Enable reverse proxy support in stlinks. Edit the stlinks.js file in the C:\Lotus\Domino\Data\domino\html\sametime\stlinks\ directory and edit the varII_RProxyName and varII_AffinityID fields, as shown from our example in Example 6-17 on page 220.

Example 6-17   stlinks.js modifications to support a reverse proxy

```
varII_RProxyName="https://bombay.itsc.austin.ibm.com"
varII_AffinityID="st"
```

5. Enable reverse proxy support in the Sametime Administrative Console:

   a. Open the Sametime Administration Tool on the Instant Messaging and Web Conferencing server by pointing your browser to the proper URL (`http://laredo.istc.austin.ibm.com/stcenter.nsf` in our environment).

   b. Click **administer the server** and log in with your Sametime administrator name.

   c. Select **Configuration** → **Connectivity**.

   d. Scroll down to the bottom of the page. In the Reverse Proxy Support section, select the **Enable Reverse Proxy Discovery on the client** setting to enable the reverse proxy support and enter the WebSEAL junction name in the Server Alias field (`st` in our example).

Restart the Instant Messaging and Web Conferencing server for the changes to take effect.

7

Integrating directory servers in an IBM WebSphere Portal environment

In this chapter, we discuss various options for using an LDAP-based directory server on an overall IBM WebSphere Portal collaboration environment. We show the process to install and configure the solution to use IBM Tivoli Directory Server, Domino, or Active Directory as the LDAP server for WebSphere Portal and Domino.

We organize this chapter into the following topics:

► IBM Tivoli Directory Server V5.2 environment

► Dual directory environment

► Microsoft Active Directory environment

7.1 IBM Tivoli Directory Server V5.2 environment

This section discusses the use of IBM Tivoli Directory Server Version 5.2 as the primary directory server for the WebSphere Portal collaborative solution. The discussion in this section is divided into the following topics:

► Installing Tivoli Directory Server V5.2
► Configuring Tivoli Directory Server

7.1.1 Installing Tivoli Directory Server V5.2

> **Note:** When installing and configuring IBM Tivoli Directory Server V5.2, we referenced the following manuals:
>
> ► *IBM Tivoli Directory Server Installation and Configuration Guide, V5.2*, SC32-1338
> ► *IBM Tivoli Directory Server Administration Guide V5.2*, SC32-1339

This section describes how to install and configure IBM Tivoli Directory Server V5.2. The high-level tasks to install the IBM Tivoli Directory Server are as follows:

► Installing IBM DB2 Universal Database

► Installing IBM GSKit V7.0.1.16

► Installing Java Runtime Environment (JRE) V1.3.1

► Installing IBM Tivoli Directory Server

► Configuring Tivoli Directory Server

Installing IBM DB2 Universal Database

IBM Tivoli Directory Server uses DB2 to store the Lightweight Directory Access Protocol (LDAP) database. This section describes how to install the IBM DB2 Universal Database V8.1, Enterprise Server Edition and supporting Fix Pack 4a.

> **Note:** We found that IBM Tivoli Directory Server V5.2, IBM Tivoli Access Manager for e-business V5.1, and IBM WebSphere Portal Extend for Multiplatforms V5.0.2 all included different versions of IBM DB2 Universal Database (UDB) V8.1. When installing DB2 UDB, it is important to understand the licensing for the product with which it is included. Information regarding the IBM Software licensing is available at:
>
> http://www.ibm.com/software/sla/sladb.nsf/
>
> We installed IBM DB2 Universal Database V8.1, Enterprise Server Edition and Fix Pack 4a (8.1.4.428).

To install DB2 Universal Database Version 8.1, Enterprise Server Edition (ESE) with Fix Pack 4a, compete the following steps:

1. Install the base DB2 UDB V8.1 using the *DB2 UDB V8.1 Enterprise Server Edition* CD. Run the **setup** command from the CD to start the installation wizard. Follow the wizard and accept most of the default values. We used the following additional information:

   – Installation Folder: C:\ibm\sqllib
   – DB2 user name: db2admin

   > **Note:** The DB2 UDB installation with the Typical installation type takes approximately 376 MB of disk space.

2. Install DB2 UDB V8.1 Fix Pack 4a for 32-bit Windows. This is the level that is officially supported by IBM Tivoli Directory Server V5.2 and WebSphere Portal V5.0.2. The contents of Fix Pack 4a can be read from:

   http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8fphist.d2w/report#WIN-32

   The DB2 UDB V8.1 Fix Pack 4a FP4a_WR21448_ESE.exe can be downloaded from:

   http://www.ibm.com/software/data/db2/udb/support/downloadv8W32fp4a.html

   After the fix pack has been installed, the **db2level** command should return 8.1.4.428.

> **Note:** If you created databases before you installed Fix Pack 4a, you will need to rebind the DB2 utilities to the databases. This step is necessary for the fixes to become effective on existing databases. The binding procedure needs to be performed only once per database. To rebind existing DB2 UDB databases after installing Fix Pack 4a, 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 GRANT PUBLIC
db2 BIND <DB2_home>\BND\@db2cli.lst GRANT PUBLIC
db2 terminate
```

Installing IBM GSKit V7.0.1.16

This section describes how to download and install IBM GSKit V7.0.1.16. GSKit is used to manage keystores and certificates. The GSKit includes the IBM Key Management utility (iKeyman) and libraries accessible to applications to create and manage certificates.

The reason we need the new GSKit is because the GSKit V7.0.1.9 installed with the Tivoli Directory Server V5.2 includes root certificates that have expired. For this reason, users will not be 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.

If you have prior version of GSKit installed, you need to check its version and uninstall it prior to installing IBM GSKit V7.0.1.16.

> **Note:** The GSKit version can be obtained by using the `gsk7ver` command or by retrieving the version from the Windows registry. We chose to use the Windows registry method, because we also needed the REGAPPS value in addition to the version.

If you have not installed Tivoli Directory Server or other software containing the IBM GSKit, you can skip this section. To determine the level of the GSKit installed, complete the following steps:

1. Start the Windows registry editor by issuing the `regedit` command.

2. Select and expand **HKEY_LOCAL_MACHINE** → **SOFTWARE** → **IBM** → **GSK7** → **CurrentVersion**.

3. Record the data value for the version name and also record the value of REGAPPS that contains the application that uses GSKit.

4. Prior to installing the new IBM GSKit V7.0.1.16, if a GSKit already has been installed, you must manually uninstall the existing IBM GSKit V7.0.1.9. The command to uninstall is `gsk7bui <LDAP>` (where <LDAP> is the name in the REGAPPS key).

> **Note:** If files still exist, such as DLLs, manually delete the C:\Program Files\IBM\GSK7 directory after the services that locked the files have been stopped.

To download and install IBM GSKit V7.0.1.16, complete the following steps:

1. The IBM GSKit V7.0.1.16 can be obtained from IBM Support or downloaded with IBM HTTP Server V1.3.28, which includes IBM GSKit V7.0.1.16, at:

   http://www.ibm.com/support/docview.wss?rs=177&context=SSEQTJ&uid=swg24006718

2. Unpack the WINDOWSPQ86671IHS1.3.28.zip file into a temporary directory and execute `gsk7bas c:\GSKtemp` to extract the GSKit installation file to the C:\GSKtemp directory.

3. Install GSKit from the C:\GSKtemp directory and execute the `setup LDAP` command, where LDAP is the application name that will be written in the REGAPPS registry value. Follow the installation wizard. We use destination folder of C:\ibm\gsk7.

Installing Java Runtime Environment (JRE) V1.3.1

In order to use the IBM Key Management utility (iKeyman) included with the GSKit, the Java Runtime Environment (JRE) V1.3.1 must be installed. The IBM Java Runtime Environment V1.3.1 is supplied in the *IBM Tivoli Access Manager V5.1 Web Administration Tool* CD in the Windows\JRE directory. Run the `install` command to start the JRE installer. We install this to C:\ibm\Java131.

We install JRE as the system JVM. To verify that the JRE is installed and available as the system JVM, verify that java.exe is found in the c:\winnt\system32 directory. You can verify the version of java.exe by executing the `java -version` command. It should return a result similar to the following:

```
java version "1.3.1"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1)
Classic VM (build 1.3.1, J2RE 1.3.1 IBM Windows 32 build cn131-20021102 (JIT
enabled: jitc))
```

Installing IBM Tivoli Directory Server

This section describes how to install IBM Tivoli Directory Server V5.2. WebSphere Portal V5.0.2 includes Tivoli Directory Server V5.1, and Tivoli Access Manager V5.1 includes Tivoli Directory Server V5.2. We choose to use Tivoli Directory Server V5.2 to avoid GSKit issues with the earlier level.

Tivoli Directory Server V5.2 includes WebSphere Application Server Express and Web Administration Tool used to manage the directory server. Tivoli Access Manager V5.1 includes the Tivoli Directory Server Web Administration Tool and Tivoli Access Manager Web Portal Manager, which can both be installed to the WebSphere Application Server provided with Tivoli Access Manager. We choose to install the Web-based administration tools on a shared WebSphere Application Server, because both the Tivoli Directory Server and Tivoli Access Manager components will be installed on the same node in our scenario.

To install Tivoli Directory Server V5.2, complete the following steps:

1. Using the *Tivoli Directory Server V5.2* CD, run **setup.exe** from the ismp folder to start the install. Follow the installation wizard.

2. The installation feature selections are shown in Figure 7-1; the following applies:

   – The destination directory is C:\ibm\ldap.
   – We preinstalled DB2 UDB and IBM GSKit.
   – We install Web Administration Tool separately from WebSphere.
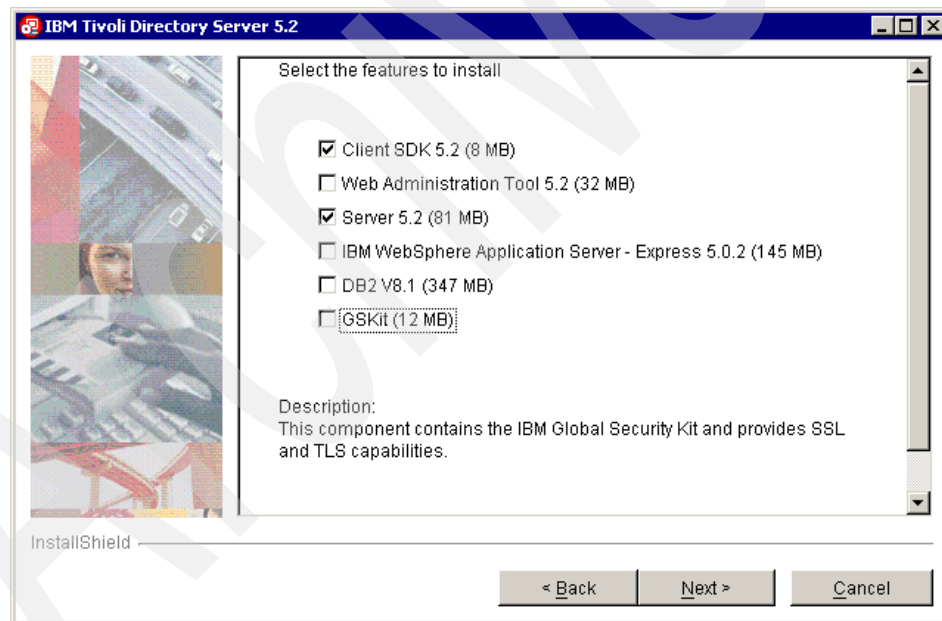
Figure 7-1   Tivoli Directory Server: Select features

7.1.2 Configuring Tivoli Directory Server

After installing Tivoli Directory Server V5.2, you need to configure the directory server.

After the first restart of your system after the Tivoli Directory Server installation, the Tivoli Directory Server Configuration Tool will be launched automatically. Alternatively, start the Directory Configuration Tool by selecting **Programs** → **Tivoli Directory Server V5.2** → **Directory Configuration**. To configure Tivoli Directory Server, complete the following steps:

1. Set the administrator distinguished name (DN) and password. Select **Administrator DN/password** under Choose a task. We use administrator DN of `cn=root`. Click **OK**.

2. Create and configure the directory database. Select **Configure database** under Choose a task.

   a. Select **Create a new database** and then click **Next**.

   b. When the Configure database - user ID window opens, enter the user ID and password created when installing DB2. We used `db2admin`. Click **Next**.

   c. When the Configure database - database name window opens, enter the database name to be created. We used `LDAPDB`. Click **Next**.

   d. When the Configure database - database code page selection opens, select **Create a universal DB2 database (UTF-8/UCS-2)** and click **Next**.

   e. When the Configure database - database location window opens, select the drive and click **Next**.

   f. When the Configuration Summary window opens, review the selections and click **Finish**.

   g. During the configuration, the configuration status will be displayed. Notice that a DB2 instance is created with the name of the user designated as the DB2 owner. When complete, click **Close**.

Note: In this stage, you can install the Web Administration Tool. For more information about this, refer to Appendix A, "Web Administration Tool for IBM Tivoli Directory Server and Tivoli Access Manager" on page 281.

7.1.3 Configuring WebSphere Portal for Tivoli Directory Server

This section describes how to configure IBM WebSphere Portal with IBM Tivoli Directory Server as the LDAP directory server. We first connect WebSphere Portal to the previously installed Tivoli Directory Server on the policy server node. In addition, we create users and groups for our sample environment.

> **Note:** For more detailed information, refer to the *WebSphere Portal V5.0.2 Information Center*:
>
> ► "Setting up IBM Directory Server," available at:
>   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_ids.html
> ► "Configuring WebSphere Portal for IBM Directory Server," available at:
>   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_ids_wp.html

The section is organized into the following tasks:

► Creating a suffix
► Defining WebSphere Portal users and groups
► Read only access for WebSphere Portal
► Configuring WebSphere Portal for security with LDAP
► Verifying the WebSphere Portal LDAP configuration

Creating a suffix

To create a suffix from Tivoli Directory Server Configuration Tool, complete the following steps on the policy server node where Tivoli Directory Server is installed:

1. Stop the IBM Tivoli Directory Server V5.2 Windows service.

2. Start the Tivoli Directory Server Configuration Tool by selecting **Programs →
   Tivoli Directory Server V5.2 → Directory Configuration**.

3. Select **Manage suffixes** under Choose a task.

4. On the suffix page, we entered o=ibm,c=us in Suffix DN for our example. Click
   Add.

5. Click **OK**.

Defining WebSphere Portal users and groups

Users and groups can be created for Tivoli Directory Server from the Web Administration Tool or by importing an LDIF file containing users and groups. We use the PortalUsers.ldif file on the *WebSphere Portal Setup* CD, as shown in Example 7-1 on page 229. We changed the DN and userpassword attribute for users wpsadmin and wpsbind to a non-trivial password.

Example 7-1   Our example WebSphere Portal LDIF file

```
version: 1

dn: o=ibm,c=us
objectclass: domain
objectclass: top
# Add lines according to this scheme that correspond to your suffix

dn: cn=users,o=ibm,c=us
objectclass: container
objectclass: top
cn: users

dn: cn=groups,o=ibm,c=us
objectclass: top
objectclass: container
cn: groups

dn: uid=wpsadmin,cn=users,o=ibm,c=us
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: wpsadmin
userpassword: passw0rd
sn: admin
givenName: wps
cn: wps admin

dn: uid=wpsbind,cn=users,o=ibm,c=us
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: wpsbind
userpassword: passw0rd
sn: bind
givenName: wps
cn: wps bind

dn: cn=wpsadmins,cn=groups,o=ibm,c=us
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=wpsadmin,cn=users,o=ibm,c=us
cn: wpsadmins
```

To import the PortalUsers.ldif file to create users and groups in Tivoli Directory Server, complete the following steps:

1. Stop the IBM Tivoli Directory Server V5.2 Windows service.

2. Start the Tivoli Directory Server Configuration Tool by selecting **Programs** → **Tivoli Directory Server V5.2** → **Directory Configuration**.

3. Select **Import LDIF data** under Choose a task.

4. When the Import LDIF Data window opens, we enter the full path of PortalUsers.ldif, select **Standard Import**, and click **Import** at the bottom of window.

5. When the import is complete, click **Close**. Close the Configuration Tool.

6. Restart the IBM Tivoli Directory Server V5.2 Windows service.

7. Verify that the LDAP entries where created properly by performing an LDAP search. For example, we entered the following from a command line window:

```
ldapsearch -h phoenix.itsc.austin.ibm.com -b o=ibm,c=us -D cn=root -w
<password> -s sub uid=wpsadmin
```

Read only access for WebSphere Portal

We chose to prevent WebSphere Portal from writing to the LDAP directory. We will only allow IBM 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. Therefore, we must configure WebSphere Portal for read-only LDAP access.

> **Note:** If you are running WebSphere Portal Versions 5.0 or 5.0.2, you will need to install PQ83389 before proceeding. This fix is already incorporated in WebSphere Portal V5.0.2 Cumulative Fix 1. More information about this fix for WebSphere Member Manager is available at:
>
> http://www.ibm.com/support/docview.wss?rs=688&context=SSHRKX&q1=PQ83389&uid=swg24006269&loc=en_US&cs=utf-8&lang=en+en

To configure WebSphere Portal for read-only LDAP access, complete the following steps:

1. Change the directory to the c:\WebSphere\PortalServer\config\templates\wmm directory on the WebSphere Portal server node.

2. We modify the following files:

   – wmm_LDAP.xml.*<LDAPType>.<number>*.wmm such as wmm_LDAP.xml.IBM_DIRECTORY_SERVER.1.wmm

– wmmLDAPAttributes_*<LDAPType>*.xml such as
wmmLDAPAttributes_IBM_DIRECTORY_SERVER.xml

Note: The following points explain the naming constructs for wmm xml files:

► <LDAPType> is 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 the lookaside database 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 7-2.

Example 7-2   Sample snippet of wmm_LDAP.xml.IBM_DIRECTORY_SERVER.1.wmm

```
<ldapRepository name="wmmLDAP"
            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. Modify the attributes in the wmmLDAPAttributes_IBM_DIRECTORY_SERVER.xml file, as shown in Example 7-3 on page 232:

– Search for wmmAttributeName=**"extId"** and modify the pluginAttributeName to ibm-entryUuid.
– Set the readOnly attribute to true in *all* attributeMap tags.

Example 7-3   Sample snippet of 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"
                        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>
```

Configuring WebSphere Portal for security with LDAP

On the WebSphere Portal server node, there are preconfigured templates that can be customized to configure WebSphere Portal for LDAP. To configure WebSphere Portal for security with LDAP, complete the following steps:

1. Open a command prompt and navigate to the <wp_home>\config directory.

2. Back up the WebSphere Portal configuration properties found in the wpconfig.properties file by entering the following command:

   `wpsconfig backup-main-cfg-file`

3. Change the wpconfig.properties values, as shown in Table 7-1 on page 233. For a detailed description of the wpconfig.properties for the LDAP security configuration with WebSphere Portal, refer to:

   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_ids_wp.html

Table 7-1   Our example wpconfig.properties values for LDAP security

Section name	Keyword	Our example value
WebSphere Application Server Properties	WasUserid	uid=wpsbind,cn=users,o=ibm,c=us
	WasPassword	<password>
Portal Configuration Properties	PortalAdminId	uid=wpsadmin,cn=users,o=ibm,c=us
	PortalAdminIdShort	wpsadmin
	PortalAdminPwd	<password>
	PortalAdminGroupId	cn=wpsadmins,cn=groups,o=ibm,c=us
	PortalAdminGroupIdShort	wpsadmins
WebSphere Portal Security LTPA and SSO Configuration	LTPAPassword	<password>
	LTPATimeout	120
	SSODomainName	.itsc.austin.ibm.com
LDAP Properties Configuration	Lookaside	false
	LDAPHostName	phoenix.itsc.austin.ibm.com
	LDAPPort	389
	LDAPAdminUId	cn=root
	LDAPAdminPwd	<password>
	LDAPServerType	IBM_DIRECTORY_SERVER
	LDAPBindID	cn=root
	LDAPBindPassword	<password>

Section name	Keyword	Our example value
Advanced LDAP Configuration	LDAPSuffix	o=ibm,c=us
	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.

5. Restart server1 and stop WebSphere_Portal application servers.

6. Change to the <wp_home>\config directory and enter the following command:

   `WPSconfig.bat validate-ldap`

   If an error occurs, review the values in the wpconfig.properties and the settings in the LDAP server. Also, ensure that the LDAP server is actually running.

7. If the validation was successful enable security by issuing the following command:

   `WPSconfig.bat enable-security-ldap`

   If the task completes successfully, you will see the message `BUILD SUCCESSFUL`.

> **Note:** You might receive the following error when running the configuration task to enable security:
>
> ```
> [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 received this error, it usually means that the following two items are true:
>
> ► Your LDAPAdminUId does not have write permissions to the LDAP server.
> ► Your WebSphere Portal server is not configured for read-only access.

8. Stop and restart the server1 application server using the wpsbind user. Start WebSphere_Portal application server.

Verifying the WebSphere Portal LDAP configuration

To verify the WebSphere Portal and LDAP configuration, complete the following steps:

1. Verify that WebSphere security is working properly by starting the WebSphere Application Server Administration Console and logging in as the user wpsbind. WebSphere security in this case provides the authentication. If security was not working, you would not be able to log in with the wpsbind user ID. In our example, we used:

   `http://pretoria.itsc.austin.ibm.com:9090/admin`

2. Verify that WebSphere Portal works 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. In our example, we used:

      `http://pretoria.itsc.austin.ibm.com/wps/portal`

      > **Important:** Using localhost or just the host name to access the portal might cause problems after configuring LDAP security. Always use the fully qualified host name for browsing.

b. From the WebSphere Portal Welcome page, click **Log in** at the top right corner (for example, we used the wpsadmin user ID and password).

7.1.4 Configuring Team Workplace with IBM Tivoli Directory Server

There are two places where you will make configuration changes to set up Lotus Team Workplace with IBM Tivoli Directory Server:

► The QuickPlace administration place

► The qpconfig.xml file

Then, you need to test the user directory.

The following sections show the configuration changes and explanations done for our example. For more detailed explanations for all of the settings in the QuickPlace administration place and qpconfig.xml file, see the *IBM Lotus Team Workplace Administrator's Guide*, available at:

http://www.lotus.com/ldd/notesua.nsf/find/quickplace

Changing the QuickPlace administration place

To change the QuickPlace administration place, complete the following steps:

1. Go to the main QuickPlace page and click **Sign in** as the QuickPlace administrator (we used http://kingston.itsc.austin.ibm.com/quickplace).

2. From the table of contents, click **Server Settings**, and then click **User Directory**. Click **Change Directory**. See Example 7-4 on page 238. Fill in the following values:

   – Type: **LDAP Server**.

   – Name: The server name of the LDAP server.

   – Port number: The default is 389, and for an SSL connection, it is 636.

   – Search base: We used user search cn=users,o=ibm,c=us.

   – Username: We used the root user to check credentials, cn=root.

   – New users: Disallow creation of new users, because this cannot be performed behind an IBM Tivoli Access Manager junction.

Figure 7-2   User directory from QuickPlace administration place

3. Click **Next**. Make sure to click **Next**, or your settings will not take effect.

> **Note:** After clicking Next, you should see your user directory along with "OK with Anonymous access," as shown in Figure 7-3 on page 238. If you see "Not OK," click **Change Directory** and correct the incorrect setting until you see "OK with Anonymous access."

Figure 7-3   Saved user directory: OK with Anonymous access

Creating the qpconfig.xml file

You will also need to enable more user directory settings for Lotus Team
Workplace to work correctly with your LDAP directory. These settings are made
in the qpconfig.xml file. To create the qpconfig.xml file, complete the following
steps:

1. Copy the qpconfig_sample.xml file from the Domino data directory.

2. Edit the qpconfig.xml file. Find the User Directory section and remove the
   following lines from the beginning and end of the <User_Directory> section,
   respectively:

   ```
   <!-- =============== START OF SAMPLE =================
   =============== END OF SAMPLE =================== -->
   ```

3. Modify the appropriate sections of this section for your user directory. The
   changes made to the our example are shown in Example 7-4.

   Important: When changing the object class, make sure that the value you
   use is the exact case as saved in your LDAP directory. For example, in our
   example, the object class for users is inetOrgPerson; setting this value to
   inetorgperson will cause problems in Team Workplace.

Example 7-4   Our example qpconfig.xml User Directory section

```
<user_directory>
    <ldap>

        <base_dn>
            <group>cn=groups,o=ibm,c=us</group>
        </base_dn>
```

```
<schema>
    <object_class>objectClass</object_class>
    <user>
        <object_class_value>inetOrgPerson</object_class_value>
        <common_name>cn</common_name>
        <display_name>cn</display_name>
        <first_name>givenname</first_name>
        <last_name>sn</last_name>
        <email>mail</email>
        <phone>telephone</phone>
    </user>
    <group>
        <object_class_value>groupOfUniqueNames</object_class_value>
        <common_name>cn</common_name>
        <display_name>cn</display_name>
        <member>uniquemember</member>
    </group>
    <dn_delimiter robust_compare="true"/>
    <dn_incoming_is_native enabled="true"/>
    <secondary_cn_component enabled="true"/>
</schema>

<search_filters>
    <authentication>
        <![CDATA[
            (|(cn={0})(uid={0})(shortname={0}))
        ]]>
    </authentication>
    <user_lookup>
        <![CDATA[
            (&(objectclass=inetOrgPerson)(sn={0})(givenname={1}))
        ]]>
    </user_lookup>
    <group_lookup>
        <![CDATA[
            (&(objectclass=groupOfUniqueNames)(cn={0}))
        ]]>
    </group_lookup>
    <group_membership>
        <![CDATA[
            (&(objectclass=groupOfUniqueNames)(uniquemember={0}))
        ]]>
    </group_membership>
</search_filters>

<member_lookup_ui>
    <column_name>
        <person>sn, givenname</person>
    </column_name>
```

```
            <column_disambiguate>
                <person>dn</person>
            </column_disambiguate>
        </member_lookup_ui>

        <search_ui_hint>
            <![CDATA[
                ( enter <B>last name, first name</B>)
            ]]>
        </search_ui_hint>
        <search_ui_index>sn</search_ui_index>

    </ldap>
</user_directory>
```

4. After these changes have been made, restart the HTTP task in Domino for Team Workplace to recognize them by issuing the following commands on the Domino console:

```
tell http q
load http
```

Testing the user directory

To make sure that the changes you made to the user directory are set correctly, you can easily test a few settings.

First, test the search functionality by signing into the QuickPlace administration place as the local QuickPlace administrator. Select **Server Settings** → **Security**. Under either Who can create new place on this server? or Who can administer this server?, click the **Add** button. Next, click the **Directory** button and search for a user and group from your LDAP directory. If an expected user or group is not returned, double check the directory settings in the Administration Console and the qpconfig.xml file as previously documented.

Second, test the authentication by signing in to the QuickPlace administration place as anyone from the LDAP directory. After you sign in, look at the source of the HTML page and search for the string .tt. You should see the following in the view source:

```
haiku.TT = 'uid=wpsadmin/cn=users/o=ibm/c=us'
```

Ensure that the DN listed is correct for your environment. If it is not, single sign-on will not work, and you need to double check the settings in the Administration Console and the qpconfig.xml file as previously documented.

7.1.5 Configuring Instant Messaging and Web Conferencing for IBM Tivoli Directory Server

There are two places where you will make configuration changes to set up Lotus Instant Messaging and Web Conferencing with IBM Tivoli Directory Server:

► The Directory Assistance database

► The Instant Messaging and Web Conferencing configuration database

Then, you need to verify the configuration changes.

Configuring the Directory Assistance database

This section provides the configuration changes and explanations for our example. For more detailed explanations for all the settings in the Directory Assistance database, see the *Lotus Domino Administrator Help*, available at:

   http://www.lotus.com/ldd/notesua.nsf/find/domino

To configure the Directory Assistance database, complete the following steps:

1. Using a Notes client open the Directory Assistance (da.nsf) database on the Instant Messaging and Web Conferencing server by selecting **File →
   Database → Open**. See Figure 7-4.
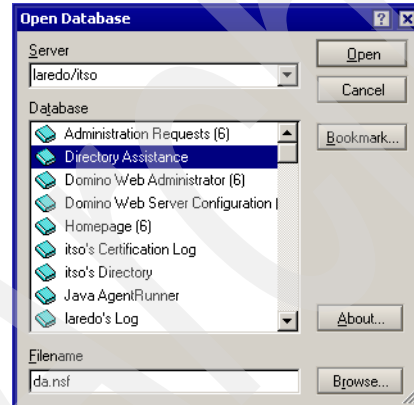
Figure 7-4   Open the da.nsf database

2. After the database is open, double-click the **LDAP** document, and click the **LDAP** tab. In the LDAP tab:

   – Set the Optional Authentication Credential: Username and Password to the bind user name and password for your directory.
   – Set the Base DN for search to the base DN for your directory.

   Figure 7-5 on page 242 shows the edited da.nsf database from our example.

Figure 7-5   The da.nsf database from our example

3. Save and close this document, and close the Directory Assistance database.

Configuring the Instant Messaging and Web Conferencing configuration database

This section provides the configuration changes and explanations for our example. For more detailed explanations for all the settings in the Instant Messaging and Web Conferencing configuration database, see the *Lotus Instant Messaging and Web Conferencing Administrator's Guide*, available at:

http://www.lotus.com/ldd/notesua.nsf/find/sametime

To configure the Instant Messaging and Web Conferencing configuration database, complete the following steps:

1. Using a Notes client, open the Instant Messaging and Web Conferencing configuration database (stconfig.nsf) on the Instant Messaging and Web Conferencing server by selecting **File** → **Database** → **Open**.

2. After the database is open, double-click the **LDAP Server** document and make the following configuration changes:

   a. Under Connection Settings:

      • Network Address of LDAP Connection: This should be your LDAP server.

- Login Name for LDAP Connection: This should be your bind user name.
- Password for LDAP Connection: This should be your bind user password.

b. Under Search Filters:

- Search filter for resolving person names: Add any attributes that you want the Instant Messaging and Web Conferencing server to search for when users are looking for other users in the directory.
- Search filter to use when resolving a user name to a distinguished name: Add any attributes that you want the users to be able to authenticate with to the Instant Messaging and Web Conferencing server.
- Search filter for resolving group names: Add any attributes that you want the Instant Messaging and Web Conferencing server to search for when users are looking for groups in the directory.

c. Under Search Base and Scope:

- Base object when searching for person entries: This should be the search base for person records from your LDAP server.
- Base object when searching for group entries: This should be the search base for group records from you LDAP server.
- The person object class used to determine if an entry is a person: `inetOrgPerson`.
- Attribute in the group object class that has the names of the group members: `uniqueMember`.
- The group object class used to determine if an entry is a group: `groupOfUniqueNames`.

Example 7-5 shows out example.

Example 7-5   Sample stconfig.nsf

```
LDAP Server Settings:

Connection Settings
        Organization Name:
        Network Address of LDAP Connection: phoenix.itsc.austin.ibm.com
        Port number for LDAP Connection:  389
        Login Name for LDAP Connection: cn=root
        Password for LDAP Connection: passw0rd
        SSL Enabled: false
        SSL Port: 636
        Search Order: 1


 Search Filters
```

```
    Search filter for resolving person names:
(&(objectclass=inetOrgPerson)(|(cn=%s*)(givenname=%s*)(sn=%s*)(mail=%s*)(uid=%s
*)))
    Search filter to use when resolving a user name to a distinguished name:
(&(objectclass=inetOrgPerson)(|(cn=%s)(givenname=%s)(sn=%s)(mail=%s)(uid=%s)))
    Search filter for resolving group names:
(&(objectclass=groupOfUniqueNames)(cn=%s*))

Search Base and Scope
    Base Objects
        Base object when searching for person entries: cn=users,o=ibm,c=us
        Base object when searching for group entries: cn=groups,o=ibm,c=us

    Scope
        Scope for searching for a person: recursive
        Scope for searching for groups: recursive

    Schema Settings

    People
        The attribute of the person entry that defines the internal ID of a
Sametime user:
        The attribute of the person entry that defines the person's name: cn
        Attribute used to distinguish between two similar person names:
        Attribute of the persion entry that defines the person's e-mail address:

        The person object class used to determine if an entry is a person:
inetOrgPerson


    Groups
        Attribute used to distinguish between two similar group names:
        The attribute of the group entry that defines the group's name: cn
        Attribute in the group object class that has the names of the group
members: uniqueMember
        The group object class used to determine if an entry is a group:
groupOfUniqueNames

    Home Server
        Name of the Home Server Attribute:
```

Note: If you make any changes to the Directory Assistance database or Instant Messaging and Web Conferencing configuration database, you will need to restart the Domino server for the changes to take effect.

Verifying the configuration changes

Point your browser to the Instant Messaging and Web Conferencing center, `http://server.domain.com/stcenter.nsf`. Click **Attend a Meeting**, and then click **Log on to Sametime**. Log in with a user name and password from the IBM Tivoli Directory Server. Does your name appear at the top right corner? If this does not work, double check the configuration settings you used in the Directory Assistance database.

Next, return to the Instant Messaging and Web Conferencing center, and click **Launch Sametime Connect**. Log in with the same user name and password. Does the connect client successfully load? If this does not work, double check the configuration settings you used with the Instant Messaging and Web Conferencing configuration database.

7.2 Dual directory environment

In some implementations where Domino LDAP has been in place, you have the choice to use IBM Tivoli Directory Server for WebSphere Portal. This requires it to perform single sign-on with a back-end systems where the authentication is performed using Domino LDAP. This implementation typically uses Lotus Team Workplace V6.5.1 and Lotus Instant Messaging and Web Conferencing V6.5.1 with Domino LDAP. This environment requires that:

- ► The Domino LDAP server must be Version 6.5.2 or later.
- ► The Name and Address book must use a V6.5.2 template design.

There are several additional steps that you might need to take to achieve this environment. You still need to enable the environment, as discussed in Chapter 3, "Implementation planning and considerations" on page 45. These additional steps are:

- ► Changing Domino LDAP and WebSphere Portal

- ► Configuring Team Workplace for a dual directory environment

- ► Configuring Instant Messaging and Web Conferencing for a dual directory environment

- ► Configuring People Finder

- ► Configuring Team Workplace to work with Instant Messaging and Web Conferencing

7.2.1 Changing Domino LDAP and WebSphere Portal

To achieve a single sign-on environment, you should perform the following configuration steps:

1. Synchronize the user directories.

   The user registries will need to by synchronized together on every Domino server in your environment, including both the Domino LDAP and mail and application servers. This synchronization is typically performed on the Domino LDAP server first, and then the changes are replicated out to other mail and application servers.

   a. From the Domino Administrative Console, open the address book from the Domino LDAP server in the Person view.

   b. Open the Person document for a user that you want to configure to enable SSO.

   c. Add the user's Tivoli Directory Server DN and login name in the User Name or Short name field. Domino requires that the levels in the LDAP name are separated by a slash (/) instead of a comma. We entered the following values:

      - User name: `uid=iuser1/cn=user/o=ibm/c=us`
      - Short name: `iuser1`
      - Domino user: `iNotes User1/ITSO`

   d. The modified iNotes User1's Person document is shown in Figure 7-6 on page 247.

   e. Ensure that the password in Tivoli Directory Server and the Internet password in the Person document match.

Figure 7-6   Person document example

2. After synchronizing the directories, you need to dereference these alias names in your Domino LDAP server:

   a. Open the Name and Address book on the Domino LDAP server.

   b. Select **Configuration** → **Messaging** → **Configurations** to open the Configuration documents view.

   c. Open the document for all server. If one does not exist, click **New document**.

   d. Select **Yes** for the Use these settings as the default settings for all servers? option.

   e. Click the **LDAP** tab and scroll to the bottom of the window. For the Allow dereferencing of aliases on search requests? option, select **Yes**, as shown in Figure 7-7 on page 248.

Figure 7-7   Dereference alias names set to Yes

3. Restart the Domino LDAP server for these changes to take effect.

7.2.2 Configuring Team Workplace for a dual directory environment

For the Lotus Team Workplace server to work in a dual directory environment, a hotfix is required from Lotus Technical Support. In our environment, we use hotfix build number 34.

This hotfix allows single sign-on to work in the WebSphere Portal environment. In a dual directory environment, the LDAP key is generated by WebSphere Portal containing the full DN of users from the WebSphere Portal LDAP server.

The Team Workplace server needs to remap the DN sent by WebSphere Portal to the DN it is expecting from the Domino Directory. Therefore, in our example, if you sign into Portal as iuser1, the following DN will be set in the LTPA token: uid=iuser1,cn=users,o=ibm,c=us. The Team Workplace server needs to read that from the token, and remap it to CN=iNotes User1,O=ibm for authentication to Team Workplace. The hotfix enables this to occur.

After you have applied the hotfix, you need to configure the Team Workplace server to work with the Team Workplace portlets. Aside from the hotfix, the configuration steps for the Team Workplace server are no different from a single

directory environment. Therefore, follow the steps in 4.4.7, "Configuring the My Team Workplace portlet" on page 101 to set up the environment correctly.

7.2.3 Configuring Instant Messaging and Web Conferencing for a dual directory environment

In a single directory environment, awareness and single sign-on rely on the LTPA token generated by WebSphere Portal. In a dual directory environment, however, this token will not work, because it will contain the full DN of the user as saved in the Portal LDAP directory and not the full DN of the Domino LDAP directory that Instant Messaging and Web Conferencing is using. Therefore, the Instant Messaging and Web Conferencing server must generate its own token, st token, to be used for awareness and single sign-on between WebSphere Portal and Instant Messaging and Web Conferencing. The following section describes how to configure WebSphere Portal and Instant Messaging and Web Conferencing to use st tokens.

Configuring WebSphere Portal to use st tokens

There are two configuration changes necessary for the Collaborative Services to use st tokens instead of the LTPA token generated by WebSphere Portal:

► Update the Collaborative Services to use st tokens.

You need to update the CSEnvironment.properties file on the WebSphere Portal server, as shown in Example 7-6. This file is located in the C:\WebSphere\PortalServer\shared\app\config in our environment.

Example 7-6   Updates to CSEnvironment.properties

```
CS_SERVER_SAMETIME_1.useLTPAToken=false
CS_SERVER_SAMETIME_1.nameFormatForResolve=loginName
CS_SERVER_SAMETIME_1.dnNameSeparator=,
```

► Allow the server application to connect to the Instant Messaging server.

To enable the Collaborative Services to communicate with the Instant Messaging server to generate the st token:

a. Open the sametime.ini file in a text editor from the Domino program directory.

b. Configure Instant Messaging and Web Conferencing to accept all IP addresses as trusted. To do this, add the following line to the Debug section at the end of the file:

```
VPS_BYPASS_TRUSTED_IPS=1
```

> **Note:** In a production environment, you can add the IP address of the
> WebSphere Portal server machine to the list of IP addresses of trusted
> servers, and remove the `[Debug]` section so that you are not accepting all
> IP addresses as trusted. Do not enter the host name. Enter the IP address.
> To do this, add the following line to the Configuration section:
>
> ```
> [Config]
> VPS_TRUSTED_IPS=trusted IP address, trusted IP address
> ```

Enabling st tokens for authentication

After configuring the Collaborative Services to generate and use st tokens, you
need to make the following configuration changes for the Instant Messaging and
Web Conferencing server to look for st tokens first, instead of the LTPA token.

You will need to make changes to the stconf.nsf and stsrc.nsf databases on the
Instant Messaging and Web Conferencing servers. The administrator who makes
these changes will need:

- ► Designer or higher access for the stconf.nsf and stsrc.nsf databases
- ► Domino Designer® Version 5.x or 6.x client
- ► Access rights to run unrestricted LotusScript/Java agents in the Server
  document's Security tab for the Instant Messaging and Web Conferencing
  server

You might want to backup the stconf.nsf, stconf30.ntf, and stsrc.nsf databases
before preceding.

To enable st tokens for authentication, complete the following steps:

1. Update the stconf.nsf database:

   a. In Domino Designer client, open the stconf.nsf database using the
      Administrator ID. Go to the forms section and select **WebAttend**.

   b. In the Objects pane, scroll to the **WebAttend (form)**.

   c. Drill down into the **WebQueryOpen**.

   d. Find the line`@Command([ToolsRunMacro];`
      `"SametimePopulateTokenLTPA");` and enter the letters `REM` in front of this
      line.

   e. Add single quotation marks around the line. Add or leave a semicolon (;)
      as the last entry on the line, as shown in Figure 7-8 on page 251.
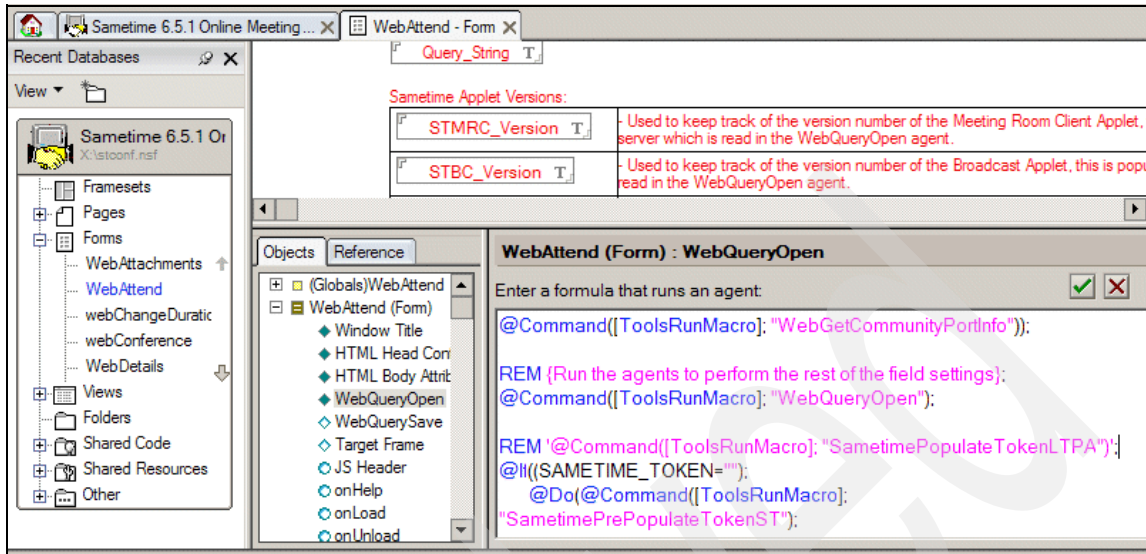
Figure 7-8   The stconf.nsf database design changes

f. Click the green check mark to verify that there are no errors.

g. Select **File** → **Database** → **Properties**.

h. Click the **Design** tab (fourth from the left).

i. Clear the **Inherit design from master template** option, as shown in Figure 7-9 on page 252.
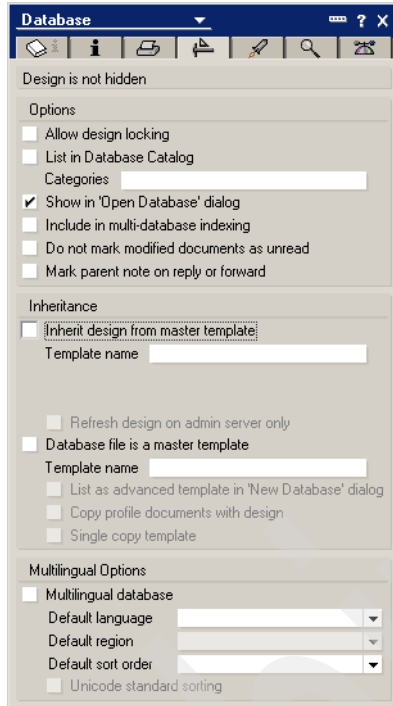
Figure 7-9   Clear the template in database properties

    j. Save and close the stconf.nsf database.

2. Update the stsrc.nsf database:

    a. In Domino Designer client, open the stsrc.nsf database using the Administrator ID. Drill down to the forms section and select **WebConnectJoin**.

    b. In the Objects pane, scroll to **WebConnectJoin(Form)**. Drill down into **WebQueryOpen**.

    c. Find the line `@Command([ToolsRunMacro];` **"**`SametimePopulateTokenLTPA`**");** and enter the letters `REM` in front of the line.

    d. Add single quotation marks around the line. Add or leave a semicolon (;) as the last entry on the line, as shown in Figure 7-10 on page 253.
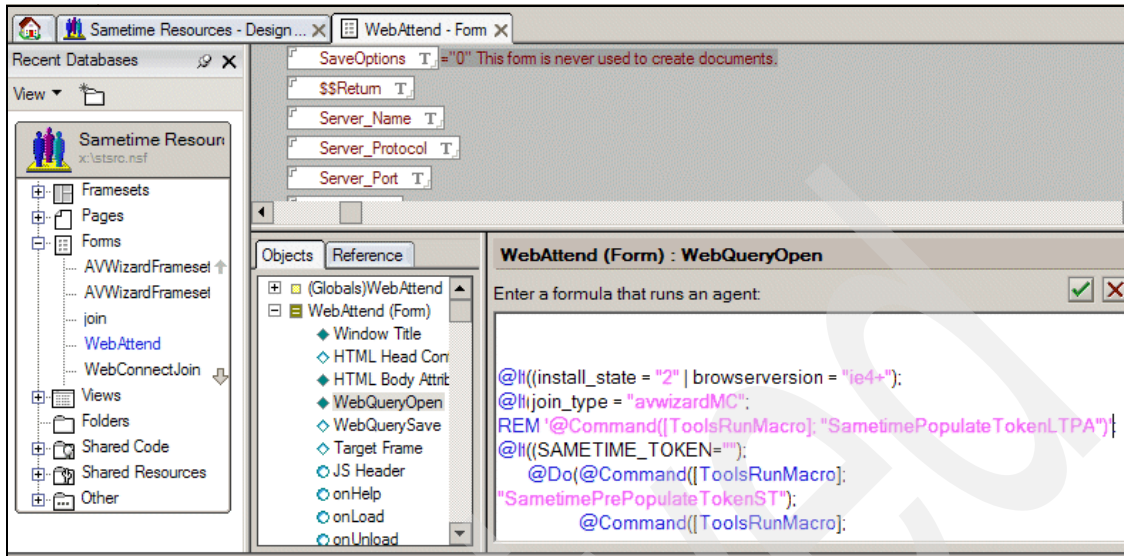
Figure 7-10   The stsrc.nsf database design changes

      e. Click the green check mark to verify that there are no errors.

      f. Save and close the database.

3. Restart the Instant Messaging and Web Conferencing server for the changes to take effect. After the Instant Messaging and Web Conferencing services start, restart WebSphere Portal.

Configuring Instant Messaging and Web Conferencing to work with the Instant Messaging portlets

After you have configured Instant Messaging and Web Conferencing and the Collaborative Services to use the st tokens generated by the Instant Messaging and Web Conferencing server, you will need to complete additional steps for the portlet to work with the Instant Messaging and Web Conferencing server. Aside from the tokens, the configuration steps for the Instant Messaging and Web Conferencing server are no different from a single directory environment. Therefore, follow the steps in 4.4.8, "Lotus Instant Messaging and Web Conferencing portlets" on page 105.

7.2.4 Configuring People Finder

People Finder will only work with the LDAP directory configured with WebSphere Portal, IBM Tivoli Directory Server in our dual directory environment. Therefore,

you must only configure People Finder to search IBM Tivoli Directory Server, as described in 4.4.12, "Configuring People Finder" on page 107.

For the Show Person Record or Show Organizational View to work from the My Team Workplaces and Lotus Web Conferencing portlets in our dual directory environment, there are two possible solutions:

► Create alias mappings between your users in the People Finder LDAP server (IBM Tivoli Directory Server in our example) and the same users in the Instant Messaging LDAP server (Domino LDAP in our example).

► Add the ibm-personAwarenessIdentity attribute to the users in your People Finder LDAP server (IBM Tivoli Directory Server in our example) to identify each user's identity in the Instant Messaging LDAP server (Domino LDAP in our example).

This section provides the instructions for adding the ibm-personAwarenessIdentity attribute.

The ibm-personAwarenessIdentity attribute places the value of a person's Distinguished Name (DN) as it appears in the LDAP directory used by Web Conferencing and Team Workplace servers into the People Finder directory on the People Finder LDAP server. The ibm-personAwarenessIdentity attribute does not enable people awareness on names that appear in the People Finder portlet. It only makes it possible to open the Person Record or Organizational View of a person whose name appears in another portlet using a different LDAP server, such as the My Lotus Team Workplaces portlet.

To use the ibm-personAwarenessIdentity attribute to extend the schema of your People Finder LDAP server, complete the following steps:

1. Use your LDAP tools to extend the LDAP schema of your People Finder LDAP server with the new object class ibm-awarenessPerson and its attribute ibm-personAwarenessIdentity.

2. Extend the object class you use for person entities (for example, inetOrgPerson, User, or whatever your People Finder LDAP server is using) with ibm-awarenessPerson as an AUX subclass. For example, to extend inetOrgPerson with this new object class, you use the rule shown in Example 7-7 on page 255.

Example 7-7   Add ibm-awarenessPerson as an AUX subclass

```
ditContentRules:
(
2.16.840.1.113730.3.2.2
NAME 'inetOrgPerson'
DESC 'Defines entries representing people in an organization's enterprise
network.'
AUX ( ibm-awarenessPerson )
)
```

This will non-destructively add the ibm-personAwarenessIdentity attribute as an optional attribute for members of the inetOrgPerson object class.

3. For each person entry in the other LDAP directory used by Web Conferencing and Team Workplace, populate the person's ibm-personAwarenessIdentity attribute with the person's Distinguished Name (DN).

   For example, if a person named iNotes User1 exists in an IBM Tivoli Directory Server directory with the Distinguished Name (DN) uid=iuser1,cn=users,o=ibm,c=us, and this person also exists in the Domino Directory as "iNotes User1" with the DN of "CN=iNotes User1,O=ibm", you need to add the ibm-personAwarenessIdentity attribute to iNotes User1's entry in the IBM Tivoli Directory Server directory with the value of the user's DN as it appears in the Domino Directory.

4. Map the ibm-personAwarenessIdentity attribute in the wmmLDAPServerAttributes.xml file on the WebSphere Portal server that hosts the People Finder portlet. You can locate the file in <wp_root>/wmm/wmmLDAPServerAttributes.xml. Add the text in Example 7-8 to the wmmLDAPServerAttributes.xml file.

Example 7-8   Text added to wmmLDAPServerAttributes.xml

```
<attributeMap
wmmAttributeName="ibm-personAwarenessIdentity"
pluginAttributeName="ibm-personAwarenessIdentity"
applicableMemberTypes="Person"
dataType="String"
valueLength="1024"
multiValued="false"/>
```

5. Map the attribute in the wmmAttributes.xml file, as shown in Example 7-9 on page 256. This file is in <wp_root>/shared/app/wmm/wmmAttributes.xml.

Example 7-9   Text added to wmmAttributes.xml

```
<attribute
wmmAttributeName="ibm-personAwarenessIdentity"
applicableMemberTypes="Person"
dataType="String"
valueLength="1024"
multiValued="true"/>
```

7.2.5 Configuring Team Workplace to work with Instant Messaging and Web Conferencing

In this configuration, the Lotus Team Workplace and Instant Messaging and Web Conferencing server are using the same directory. Configuring these servers to work together does not change if they are both configured with IBM Tivoli Directory Server, or both configured with Domino LDAP. Therefore, you should follow the steps provided in 4.4.6, "Lotus Team Workplace portlets settings" on page 100, to correctly configure Team Workplace to work with Instant Messaging and Web Conferencing.

7.3 Microsoft Active Directory environment

In this section, we discuss the use of Microsoft Active Directory as the LDAP server that will be used in the collaborative portal environment. This scenario is typically used when Microsoft Active Directory has been deployed as the primary directory provider in the enterprise. In this discussion, we assume that Microsoft Active Directory has been configured properly, and we only discuss the WebSphere Portal and Domino Extended Products configuration for this environment.

This section describes the procedure used to install and configure the WebSphere Portal server node for our example runtime environment on Microsoft Windows. It is divided into the following high-level topics:

► WebSphere Portal and Microsoft Active Directory

► Configuring single sign-on

► Configuring Team Workplace with Microsoft Active Directory

► Configuring Instant Messaging and Web Conferencing for Microsoft Active Directory

► Configuring People Finder for Microsoft Active Directory

► Configuring Tivoli Access Manager

7.3.1 WebSphere Portal and Microsoft Active Directory

The basic setup for WebSphere Portal is discussed in 4.2, "Implementing IBM WebSphere Portal" on page 59. There are additional things that you should do to use Microsoft Active Directory for the directory provider with WebSphere Portal. You need to configure Microsoft Active Directory for additional attributes and set up WebSphere Portal to connect to Microsoft Active Directory.

Defining the preferredLanguage attribute

The preferredLanguage attribute must be added to the Active Directory scheme to create and modify portal users. To add the preferredLanguage attribute to Active Directory, complete the following steps:

1. If you have not installed the Windows 2000 Support Tool, install it from the \Support\tools directory on the *Windows 2000 Setup* CD.

2. Add the Active Directory Schema Snap-in using the following steps:

   a. Register the schmmgmt.dll file using the `regsvr32 schmmgmt.dll` command.

   b. From the Windows Start menu, select **Programs** → **Windows 2000 Support Tools** → **Tools** → **Security Administration Tools**.

   c. Select **Console** → **Add/Remove Snap-in**.

   d. From the Standalone tab, click **Add**.

   e. Select **Active Directory Schema** from the list of Available Standalone Snap-ins. Click **Add** to add the snap-in to the console.

   f. Click **Close**.

   g. Click **OK** to return to the Security Administration Tools console.

3. Configure the Active Directory Schema Snap-in using the following steps:

   a. From the Security Administration Tools console, right-click **Active Directory Schema** and select **Operations Master**.

   b. Select the **The Schema can be modified on this Domain Controller** option.

   c. Click **OK** to save this change.

4. Create the preferredLanguage attribute using the following steps:

   a. From the Security Administration Tools console, expand **Active Directory Schema** → **Attributes** → **Create Attribute**.

   b. Click **Continue** to access the new attribute properties.

   c. Enter the following values for new attributes:

   - Common Name: `preferredLanguage`

- LDAP Display Name: `preferredLanguage`
- X500 Object ID: `2.16.840.1.113730.3.1.39`
- Syntax: `Case Insensitive String`

    d. Click **OK** to create the preferredLanguage attribute.

5. Add the preferredLanguage attribute to the user object class using the following steps:

    a. From the Security Administration Tools console, expand **Active Directory Schema** → **Classes**.

    b. Right-click **user** and chose **Properties** to open the user properties.

    c. Select the **Attributes** tab, and in the Optional section, click **Add** to add a new schema object.

    d. Select **preferredLanguage** from the list of objects, and click **OK** to add this object.

    e. Click **OK** to return to the Security Administration Tools console.

    f. Close the Security Administration Tool.

6. Create new users for the WebSphere administrator and Portal administrator. These user names should not contain any spaces. We choose to use wpsbind for WebSphere administrator and wpsadmin for Portal administrator. Because we are using the domain name itsc.austin.ibm.com for our machines, the full canonical names for these users are:

    – CN=wpsbind,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
    – CN=wpsadmin,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com

7. Create a group for Portal administrator groups. This name also should not contain any spaces. We choose to call this group wpsadmins. The full name of this group is
CN=wpsadmins,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com.

Now, the Microsoft Active Directory is ready to be used by WebSphere Portal. You need to add the preferredLanguage mapping to the Member Manager XML file using the following steps:

1. Use a text editor to open the
<wp_root>\PortalServer\config\templates\wmm\wmmLDAPAttributes_ACTIVE_DIRECTORY.xml file.

2. Add the following attribute map tag:

```
<attributeMap    wmmAttributeName="preferredLanguage"
                 pluginAttributeName="preferredLanguage"
                 applicableMemberTypes="Person"
                 dataType="String"
                 valueLength="128"
                 multiValued="false" />
```

3. Save and close the text file.

Configuring LDAP security with Microsoft Active Directory

On the WebSphere Portal server node, there are preconfigured templates that must be customized to use the Microsoft Active Directory LDAP. The following customization is performed on the WebSphere Portal server machine:

1. Configure WebSphere Portal to use Microsoft Active Directory. This task is similar to the steps in "Configuring WebSphere Portal for security with LDAP" on page 232. Open a command prompt and navigate to the C:\WebSphere\PortalServer\config directory.

2. Back up the WebSphere Portal configuration properties in the wpconfig.properties file by entering the following command:

   wpsconfig backup-main-cfg-file

3. Change the wpconfig.properties values, as shown in Table 7-2. For a detailed description of the wpconfig.properties values for LDAP security configuration with WebSphere Portal, refer to the *WebSphere Portal Information Center* (search Configuring WebSphere Portal for Active Directory), available at:

   http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/

Table 7-2   Our sample wpconfig.properties values for Active Directory security

Section	Keyword	Our example value
WebSphere Application Server properties	WasUserid	CN=wpsbind,CN=Users,DC=itsc,DC=austin, DC=ibm,DC=com
	WasPassword	wpsbind
Portal configuration properties	PortalAdminId	CN=wpsadmin,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
	PortalAdminPwd	wpsadmin
	PortalAdminIdShort	wpsadmin
	PortalAdminGroupId	CN=wpsadmins,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
	PortalAdminGroupIdShort	wpsadmins

Section	Keyword	Our example value
WebSphere Portal Security LTPA and SSO configuration	SSOEnabled	true
	LTPAPassword	<password>
	LTPATimeout	120
	SSODomainName	.itsc.austin.ibm.com
	SSORequiresSSL	false
LDAP Properties Configuration	Lookaside	False
	LDAPHostName	warsaw.itsc.austin.ibm.com
	LDAPPort	389
	LDAPAdminUid	CN=administrator,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
	LDAPAdminPwd	<password>
	LDAPServerType	ACTIVE_DIRECTORY
	LDAPBindID	CN=administrator,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
	LDAPBindPassword	<password>
Advanced LDAP Configuration	LDAPUserFilter	(&(l(cn=%v)(samAccountName=%v))(objectclass=person))
	LDAPGroupFilter	(&(cn=%v)(objectclass=group))
	LDAPSuffix	DC=itsc,DC=austin,DC=ibm,DC=com
	LdapUserPrefix	cn
	LdapUserSuffix	CN=Users
	LdapGroupPrefix	cn
	LdapGroupSuffix	CN=Users
	LDAPUserObjectClass	person
	LDAPGroupObjectClass	group
	LDAPGroupMember	member
	LDAPsslEnabled	False

Section	Keyword	Our example value
Credentials for WebSphere Application Server administration secure SOAP connection	TrustStore	/etc/DummyClientTrustFile.jks
	TrustStorePwd	WebAS
	KeyStore	/etc/DummyClientKeyFile.jks
	KeyStorePwd	WebAS

4. Save the updated wpconfig.properties file.

5. Start the server1 application server and stop the WebSphere_Portal application server using the following commands:

```
startServer server1
stopServer WebSphere_Portal
```

6. Change to the <wp_home>\config directory and enter the following command:

```
WPSconfig.bat validate-ldap
```

If an error occurs, review the values in wpconfig.properties (typographical errors are quite often the cause of an error for this step) and the settings in the LDAP server. Also, ensure that the LDAP server is running.

7. If the validation was successful, enable security by issuing the following command:

```
WPSconfig.bat enable-security-ldap
```

If the task completes successfully, you will see the message BUILD SUCCESSFUL.

8. Restart the server1 application server to activate the configuration and use wpsbind to stop the server.

Verifying the LDAP configuration

To verify the WebSphere Portal and LDAP configuration, complete the following steps:

1. Verify that WebSphere security is working properly by starting the WebSphere Application Server Administration Console and logging in as the user ID wpsbind. WebSphere security in this case provides the authentication. If security was not working, you would not be able to log in with the wpsbind user ID. In our example, we used:

```
http://pretoria.itsc.austin.ibm.com:9090/admin
```

2. Verify that WebSphere Portal works properly with the LDAP configuration and WebSphere security:

a. If everything works properly, you should be able to browse to your WebSphere Portal server using the fully qualified host name, which is now configured to use LDAP. In our example, we used:

```
http://pretoria.itsc.austin.ibm.com/wps/portal
```

> **Important:** Using localhost or just the host name for accessing the portal might cause problems after configuring LDAP security. Always use the fully qualified host name for browsing.

b. From the WebSphere Portal Welcome page, click **Log in** at the top right corner (for example, we used the wpsadmin user ID and password).

7.3.2 Configuring single sign-on

Now that the WebSphere Portal has been setup to use Microsoft Active Directory, the Domino LDAP still needs to be used for the Mail and QuickPlace portlets, as discussed in 7.2.1, "Changing Domino LDAP and WebSphere Portal" on page 246.

To configure single sign-on, complete the following steps:

1. Synchronize the user directories.

The user registries will need to be synchronized together on every Domino server in your environment, including both the Domino LDAP and mail and application servers. This synchronization is typically performed on the Domino LDAP server first, and then the changes are replicated out to the other mail and application servers:

a. From the Domino Administrative Console, open the address book from the Domino LDAP server in the Person view.

b. Open the Person document for a user you want to configure to enable SSO.

c. Add the user's IBM Tivoli Directory Server DN and login name in the User Name or Short name field. Domino requires that the levels in the LDAP name are separated by a slash (/) instead of a comma. We used:

- User name:
  `CN=iNotes User1,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com`
- Short name: `iuser1`
- Domino user: `iNotes User1/ITSO`

d. The modified iNotes User1's Person document is shown in Figure 7-11 on page 263.

e. Ensure that the password in IBM Tivoli Directory Server and the Internet password in the Person document match.

> **Important:** In our example, the Active Directory DN is added as the last line of the User name field, and the Portal login ID (sAMAccountName) is added to the Short name field. You can add these values anywhere you like in either field, as long as they are not the top line of the User name field. This must remain the Domino canonical name.

Figure 7-11   Person document example

Testing single sign-on to Domino

The following steps explain how to properly test single sign-on between WebSphere Portal and your Domino mail or application server:

1. Sign on to WebSphere Portal.

2. Make sure that you have a Domino database where the access control list (ACL) has the **-Default-** and **Anonymous** access set to **No Access**. In our example, we tested with iNotes User1's mail file.

3. Change the URL in the browser to the protected file. In our example:

    ```
    http://kingston.istc.austin.ibm.com/mail/iuser1.nsf
    ```

If you are prompted to enter your user name and password, SSO does not work between WebSphere Portal and Domino. The Technote *Troubleshooting*

WebSphere Portal, Sametime, QuickPlace and Domino SSO Issues, 1158269, should help further troubleshoot your single sign-on problems. This Technote is available at:

```
http://www.ibm.com/support/docview.wss?rs=0&uid=swg21158269
```

Changing the LDAP canonical name

Our IBM Tivoli Directory Server name is cn=wpsadmin,o=ibm,c=us.

If you want to add the user wpsadmin, you would add the following to the field: CN=wpsadmin/CN=Users/DC=itsc/DC=austin/DC=ibm/DC=com

To add all members in the /DC=itsc/DC=austin/DC=ibm/DC=com organization add the following to the field: */DC=itsc/DC=austin/DC=ibm/DC=com

The group cn=wpsadmins would not work in our example, because that group is not located in the Domino Directory.

7.3.3 Configuring Team Workplace with Microsoft Active Directory

There are two places where you will make configuration changes to set up Lotus Team Workplace with the Microsoft Active Directory:

► The QuickPlace administration place

► The qpconfig.xml file

Then, you need to test the user directory.

The following sections provide the configuration changes and explanations for our example. For more detailed explanations for all of the settings in the QuickPlace administration place and the qpconfig.xml file, see the *Lotus Team Workplace Administrator's Guide*, available at:

```
http://www.lotus.com/ldd/notesua.nsf/find/quickplace
```

Changing the QuickPlace administration place

To change the QuickPlace administration place, complete the following steps:

1. Access the QuickPlace administration using a Web browser. We used the following URL:

   ```
   http://kingston.itsc.austin.ibm.com/quickplace
   ```

2. Click **Sign In** and use the administrator user name and password that you specified during the installation (in our case, qpadmin).

3. From the table of contents, select **Server Settings** → **User Directory** → **Change Directory**. Our settings are shown in Figure 7-12 on page 265.

Figure 7-12   User directory from the QuickPlace administration place

Our field values are:

- – Type: **LDAP Server**

- – Name: Microsoft Active Directory server name:
  `warsaw.itsc.austin.ibm.com`

- – Port number: The port number on which the LDAP server listens. The
  default is `389`; for SSL, it is `636`.

– Search base: This is the search base for users and groups:
  `CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com`

– LDAP user name: `CN=Administrator,CN=Users,DC=itsc,DC=austin,`
  `DC=ibm,DC=com`

4. We do not want place managers to be able to register new users, because we use IBM Tivoli Access Manager, and QuickPlace is put behind a secure junction.

5. Click **Next**. Make sure to click Next, or your settings will not take effect. After clicking Next, you should see your user directory along with the "OK with Anonymous access" message, as shown in Figure 7-13. If you see "Not OK," click **Change Directory** and correct the incorrect setting until you see "OK with Anonymous access."
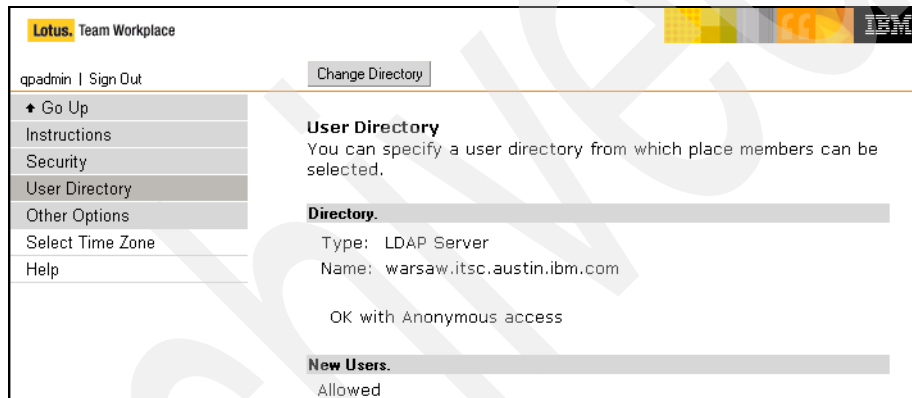
Figure 7-13   Saved user directory: OK with Anonymous access

Creating the qpconfig.xml file

You will also need to enable more user directory settings for Team Workplace to work correctly with your LDAP directory. These settings are made in the qpconfig.xml file. To create the qpconfig.xml file, complete the following steps:

1. Copy the qpconfig_sample.xml file from the Domino data directory.

2. Edit the qpconfig.xml file. Find the User Directory section and remove the following lines from the beginning and end of the <User_Directory> section, respectively:

```
<!-- =============== START OF SAMPLE =================
=============== END OF SAMPLE =================== -->
```

3. Modify the appropriate sections of this section for your user directory. The changes made to our example are shown in Example 7-10 on page 267.

> **Important:** When changing the object class, make sure that the value you use is the exact case as saved in your LDAP directory. For example, in our example, the object class for users is inetOrgPerson; setting this value to inetorgperson will cause problems in Team Workplace.

Example 7-10   User Directory section from qpconfig.xml

```
<user_directory>
      <ldap>

          <base_dn>
             <group>CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com</group>
          </base_dn>

          <schema>
             <object_class>objectClass</object_class>
             <user>
                <object_class_value>person</object_class_value>
                <common_name>cn</common_name>
                <display_name>cn</display_name>
                <first_name>givenname</first_name>
                <last_name>sn</last_name>
                <email>mail</email>
                <phone>telephone</phone>
             </user>
             <group>
                <object_class_value>groupOfNames</object_class_value>
                <common_name>cn</common_name>
                <display_name>cn</display_name>
                <member>member</member>
             </group>
             <dn_delimiter robust_compare="true"/>
             <dn_incoming_is_native enabled="true"/>
             <secondary_cn_component enabled="true"/>
          </schema>

          <search_filters>
             <authentication>
                <![CDATA[
                   (|(cn={0})(sAMAccountName={0}))
                ]]>
             </authentication>
             <user_lookup>
                <![CDATA[
                   (&(objectclass=person)(sn={0})(givenname={1}))
                ]]>
             </user_lookup>
```

```
        <group_lookup>
           <![CDATA[
              (&(objectclass=group)(cn={0}))
           ]]>
        </group_lookup>
        <group_membership>
           <![CDATA[
              (&(objectclass=group)(member={0}))
           ]]>
        </group_membership>
     </search_filters>

     <member_lookup_ui>
        <column_name>
           <person>sn, givenname</person>
        </column_name>
        <column_disambiguate>
           <person>dn</person>
        </column_disambiguate>
     </member_lookup_ui>

     <search_ui_hint>
        <![CDATA[
           ( enter <B>last name, first name</B>)
        ]]>
     </search_ui_hint>
     <search_ui_index>sn</search_ui_index>

  </ldap>
</user_directory>
```

4. After these changes have been made, restart the HTTP task in Domino for Team Workplace to recognize them by issuing the following commands on the Domino console:

```
tell http q
load http
```

Testing the user directory

To make sure that the changes you made to the user directory are set correctly, you can easily test a few settings.

First, test the search functionality by signing in to the QuickPlace administration place as the local QuickPlace administrator. Select **Server Settings → Security**. Under either the Who can create new place on this server? or Who can administer this server? option, click the **Add** button. Next, click the **Directory** button and search for a user and group from your LDAP directory. If an expected

user or group is not returned, double check the directory settings in the administration console and the qpconfig.xml file previously documented.

Second, test authentication by signing in to the QuickPlace administration place as anyone from the LDAP directory. After you have signed in, look at the source of the HTML page and search for the string `.tt`. You should see the following in the view source:

```
haiku.TT = "CN=Administrator/CN=Users/DC=itsc/DC=ibm/DC=com"
```

Make sure that the DN listed is correct for your environment. If it is not, single sign-on will not work, and you need to double check the settings in the administration console and the qpconfig.xml file previously documented.

7.3.4 Configuring Instant Messaging and Web Conferencing for Microsoft Active Directory

There are two places where you will make configuration changes to set up Lotus Instant Messaging and Web Conferencing with Microsoft Active Directory:

► The Directory Assistance database

► The Instant Messaging and Web Conferencing configuration database

Then, you need to test the configuration changes.

Configuring the Directory Assistance database

This section provides the configuration changes and explanations for our example. For more detailed explanations for all of the settings in the Directory Assistance database, see the *Lotus Domino Administrator Help*, available at:

http://www.lotus.com/ldd/notesua.nsf/find/domino

To configure the Directory Assistance database, complete the following steps:

1. Using a Notes client, open the Directory Assistance (da.nsf) database on the Instant Messaging and Web Conferencing server by selecting **File →
Database → Open**. See Figure 7-14 on page 270.
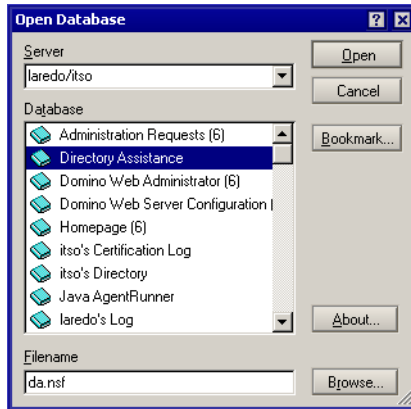
Figure 7-14   Open the da.nsf database

2. After the database is open, double-click the **LDAP** document, and click the **LDAP** tab. In the LDAP tab, we used the following values:

   – Set the Optional Authentication Credential: Username and Password to the bind user name and password for your directory.
   – Set the Base DN for search to:
     `CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com`
   – Set the Type of search filter to `Active Directory`.

   The edited da.nsf database from our example is shown in Figure 7-15 on page 271.

DIRECTORY ASSISTANCE

Basics | Naming Contexts (Rules) | LDAP |

LDAP Configuration

Hostname:	warsaw.itsc.austin.ibm.com
Optional Authentication Credential:	CN=Administrator,CN=Users,DC=itsc,DC=austin,DC
Username:	=ibm,DC=com
Password:	swj43r
Base DN for search:	CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
Channel encryption:	None
Port:	389

Advanced Options

Timeout:	60 seconds
Maximum number of entries returned:	100
Dereference alias on search:	Always
Preferred mail format:	Internet Mail Address
Attribute to be used as Notes Distinguished Name:	
Type of search filter to use:	Active Directory

Figure 7-15   The da.nsf database from our example

3. Save and close this document, and close the Directory Assistance database.

Configuring the Instant Messaging and Web Conferencing configuration database

This section provides the configuration changes and explanations for our example. For more detailed explanations for all of the settings in the Instant Messaging and Web Conferencing configuration database, see the *Lotus Instant Messaging and Web Conferencing Administrator's Guide*, available at:

http://www.lotus.com/ldd/notesua.nsf/find/sametime

To configure the Instant Messaging and Web Conferencing configuration database, complete the following steps:

1. Using a Notes client, open the Instant Messaging and Web Conferencing configuration (stconfig.nsf) database on the Instant Messaging and Web Conferencing server by selecting **File** → **Database** → **Open**.

2. After the database is open, double-click the **LDAP Server** document and make the following configuration changes:

   a. Under Connection Settings:

      • Network Address of LDAP Connection: This should be your LDAP server.

- Login Name for LDAP Connection: This should be your bind user name.
- Password for LDAP Connection: This should be your bind user password.

b. Under Search Filters:

- Search filter for resolving person names: Add any attributes that you want the Instant Messaging and Web Conferencing server to search for when users are looking for other users in the directory.
- Search filter to use when resolving a user name to a distinguished name: Add any attributes that you want the users to be able to authenticate with to the Instant Messaging and Web Conferencing server.
- Search filter for resolving group names: Add any attributes that you want the Instant Messaging and Web Conferencing server to search for when users are looking for groups in the directory.

c. Under Search Base and Scope:

- Base object when searching for person entries: This should be the search base for person records from your LDAP server.
- Base object when searching for group entries: This should be the search base for group records from you LDAP server.
- The person object class used to determine if an entry is a person: `person`.
- Attribute in the group object class that has the names of the group members: `member`.
- The group object class used to determine if an entry is a group: `group`.

Our example is shown in Example 7-11.

Example 7-11   stconfig.nsf from our example

```
LDAP Server Settings:

Connection Settings
        Organization Name:
        Network Address of LDAP Connection: warsaw.itsc.austin.ibm.com
        Port number for LDAP Connection:   389
        Login Name for LDAP Connection:
CN=Administrator,CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
        Password for LDAP Connection: <password>
        SSL Enabled: false
        SSL Port: 636
Search Order: 1

 Search Filters
```

```
      Search filter for resolving person names:
(&(objectclass=person)(|(cn=%s*)(givenname=%s*)(sn=%s*)(mail=%s*)(sAMAccountNam
e=%s*))))
      Search filter to use when resolving a user name to a distinguished name:
(&(objectclass=person)(|(cn=%s)(givenname=%s)(sn=%s)(mail=%s)(sAMAccountName=%s
)))
      Search filter for resolving group names: (&(objectclass=group)(cn=%s*))

Search Base and Scope
    Base Objects
        Base object when searching for person entries:
CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com
        Base object when searching for group entries:
CN=Users,DC=itsc,DC=austin,DC=ibm,DC=com

    Scope
        Scope for searching for a person: recursive
        Scope for searching for groups: recursive

    Schema Settings

    People
        The attribute of the person entry that defines the internal ID of a
Sametime user:
        The attribute of the person entry that defines the person's name: cn
        Attribute used to distinguish between two similar person names:
        Attribute of the person entry that defines the person's e-mail address:
        The person object class used to determine if an entry is a person:

person


    Groups
        Attribute used to distinguish between two similar group names:
        The attribute of the group entry that defines the group's name: cn
        Attribute in the group object class that has the names of the group
members: member
        The group object class used to determine if an entry is a group: group

    Home Server
        Name of the Home Server Attribute:
```

Note: If you make any changes to the Directory Assistance database or Instant Messaging and Web Conferencing configuration database, you will need to restart the Domino server for the changes to take effect.

Testing the configuration changes

Point a browser to the Instant Messaging and Web Conferencing center, `http://server.domain.com/stcenter.nsf`. Click **Attend a meeting**, and then click **Log on to Sametime**. Log in with a user name and password from the IBM Directory server. Does your name appear at the top right corner? If this does not work, double check the configuration settings you used in the Directory Assistance database.

Next, go back to the Instant Messaging and Web Conferencing center, and click **Launch Sametime Connect**. Log in with the same user name and password. Does the connect client successfully load? If this does not work, double check the configuration settings you used with the Instant Messaging and Web Conferencing configuration database.

7.3.5 Configuring People Finder for Microsoft Active Directory

Depending on the configuration of your LDAP directory, when People Finder is placed on a page, it will display the following instructions:

```
People Finder is an online corporate directory that lets users search for and
view a colleague's contact information and position in the organization's
structure.
To configure People Finder, you must:
Verify that WebSphere Member Manager (WMM) has been installed and configured
correctly.
Click the Configure (wrench) icon above to select the fields to show in search
results and person records.
```

To resolve this message, complete the following steps:

1. Sign in to Portal as the Portal administrator (wpsadmin in our example).

2. Click the configure (wrench) icon.

3. Here, you will see a list of fields in red that are causing the problem. In our example, the fields that are causing the error are employeeNumber, localityName, pager, departmentNumber, secretary, countryName, preferredLanguage, roomNumber, businessCategory, labeledURI, employeeType, stateOrProvinceName, and ibm-personalTitle. This can be seen in Figure 7-16 on page 275.
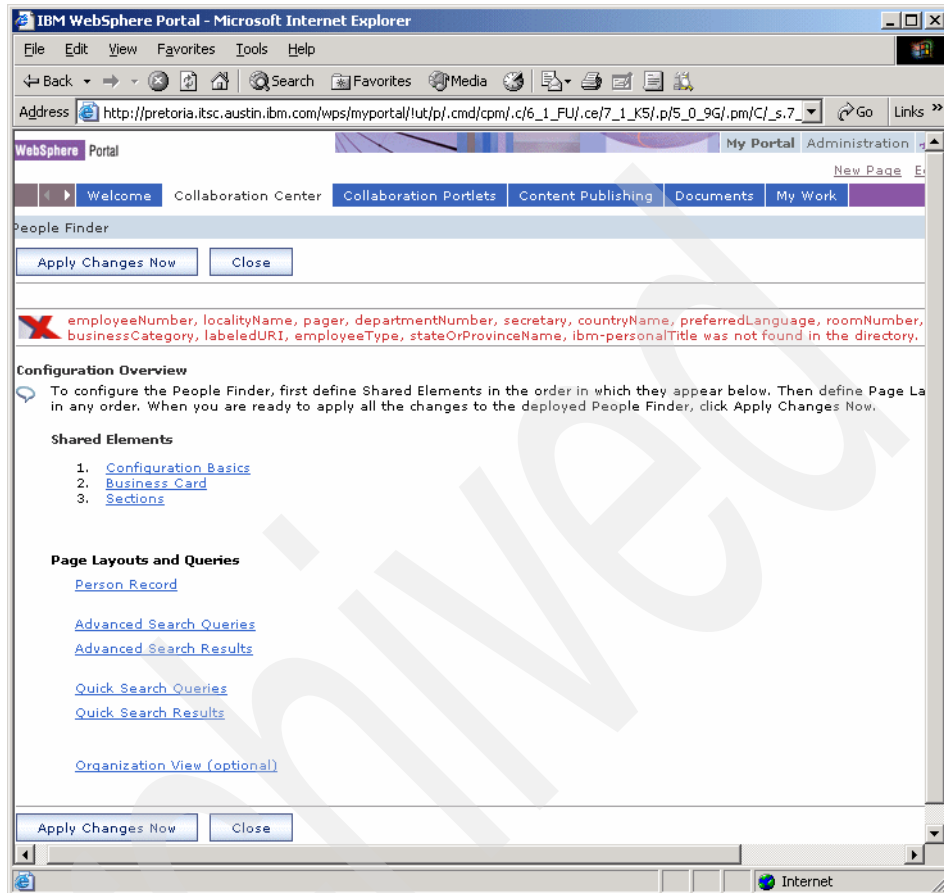
Figure 7-16   Fields not found in the directory

4. You need to look in every section of the portlet for references to this field and remove them. As soon as the field has been completely removed, it will no longer appear in the list.

5. In our example, the fields were found in the location listed in Table 7-3 on page 276.

6. Click the trash can icon to remove the field from this section, and the field will disappear if this is the only place it appears.

7. If it disappears, click **OK** to save the changes in that section, and then click **Apply Changes Now** to return to the People Finder page. At this point, the portlet should give you search options to find people.

8. If all fields do not disappear, continue to go through each section until they are gone. Also, two of the links, Person Record and Advanced Search Queries, contain multiple sections under the Select a Section heading. If you remove a field from one section, click **OK**, and then **Apply Changes Now**. Click the configuration (wrench) icon again to explore the additional sections in each link.

Table 7-3   Fields not found locations table

Field	Page layouts and queries	Section (if applicable)
employeeNumber	Person Record	Contact Information
	Advanced Search Queries	Contact Information
localityName	Business Card	
	Advanced Search Queries	Contact Information
	Person Record	Contact Information
pager	Person Record	Contact Information
	Advanced Search Queries	Contact Information
departmentNumber	Person Record	Current Job
	Advanced Search Queries	Current Job
secretary	Person Record	Contact Information
countryName	Person Record	Contact Information
	Advanced Search Queries	Contact Information
preferredLanguage	Person Record	Background
	Advanced Search Queries	Background
roomNumber	Person Record	Contact Information
	Advanced Search Queries	Contact Information
businessCategory	Person Record	Current Job
labeledURI	Person Record	Background
employeeType	Person Record	Current Job
	Advanced Search Queries	Current Job
stateOrProvinceName	Advanced Search Queries	Contact Information
	Person Record	Contact Information
ibm-personalTitle	Person Record	Contact Information

Show Person Record in menu

The Show Person Record menu item, as shown in Figure 7-17, does not appear by default in the People Finder portlet for all users, only for the Portal administrator.
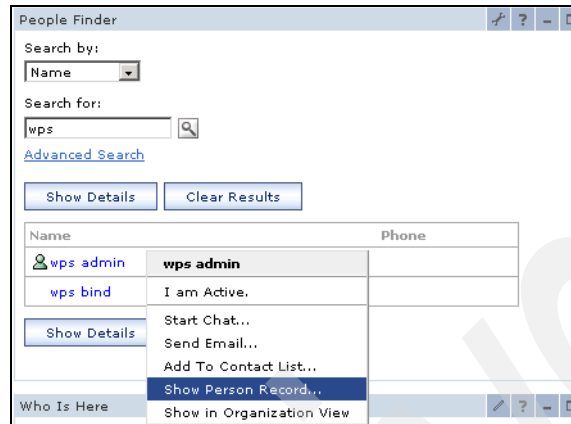
Figure 7-17   Show Person Record menu item

To enable this to appear for all users, you must give all users access to the hidden page for the People Finder as follows:

1. Sign in to Portal as the portal administrator.

2. Select **Administration** → **Portal User Interface** → **Manage Pages**.

3. Under Context Root, find the lotus.workplace.hidden.page.PeopleFinder page.

4. Give All authenticated portal users user access to this page. The users will not see any additional pages in Portal, but should now see the menu option Show Person Record.

If you continue to experience problems with the Show Person Record option, see the Technote *People Finder Portlet Does not Display 'Show Person Record' Menu Choice*, 1174638, to continue troubleshooting. This is available at:

    http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174638

7.3.6 Configuring Tivoli Access Manager

For a complete discussion about the configuration of IBM Tivoli Access Manager, see 6.2, "Installing the policy server node" on page 169. Specific instructions that apply to using Microsoft Active Directory as the LDAP server for configuring Access Manager Runtime is discussed in "Configuring Tivoli Access Manager runtime" on page 172.

You need to configure the following:

- ► Specify to use Active Directory.
- ► On the Active Directory Server information window, we used the following values:
  - Configure to Multiple Active Directory Domains: **No**
  - Active Directory host name: `warsaw.itsc.austin.ibm.com`
  - Active Directory Domain: `DC=itsc,DC=austin,DC=ibm,DC=com`
  - Enable Encrypted Connection: **No**
- ► On the Active Directory Administrator information window, use Administrator and its password.
- ► When the Active Directory data information window opens, enter the distinguished name. We entered:

  `DC=itsc,DC=austin,DC=ibm,DC=com`

As the Domain administrator, you must add the policy server node and the reverse proxy node to the Microsoft Active Directory domain.

In the reverse proxy server, when configuring WebSEAL, these are the options that you need to provide for the Microsoft Active Directory. These selections are from the Access Manager Configuration utility, when selecting the Access Manager Runtime package:

1. When the Access Manager Policy Server Host window opens, select **Access Manager Policy Server is installed on another machine**. Enter the appropriate values and then click **Next**. We entered:

   - Host name: `warsaw.itsc.austin.ibm.com`
   - Listening port: `7135`

2. When the Registry window opens, select **Active Directory** and click **Next**.

3. When the Active Directory Server Information window opens, we entered the following values and then clicked **Next**:

   - Configure to Multiple Active Directory Domains: **No**
   - Active Directory host name: `warsaw.itsc.austin.ibm.com`
   - Active Directory Domain: `DC=itsc,DC=austin,DC=ibm,DC=com`
   - Enable Encrypted Connection: **No**

4. When the Active Directory administrator information window opens, we entered the following values and then clicked **Next**:

   - Administrator ID: `administrator`
   - Password: `<password>`

5. When the Active Directory data information window opens, we entered the following value and then clicked **Next**:

   – Distinguished name: `DC=itsc,DC=austin,DC=ibm,DC=com`

6. When the LDAP Server Information window opens, we entered the following values and then clicked **Next**:

   – LDAP host name: `warsaw.itsc.austin.ibm.com`
   – LDAP port number: 389

Web Administration Tool for IBM Tivoli Directory Server and Tivoli Access Manager

This appendix lists the steps to implement the IBM Tivoli Web Administration Tool for IBM Tivoli Directory Server and IBM Tivoli Access Manager. We divide this appendix into the following topics:

► Installing Tivoli Web Administration Tool overview

► Installing WebSphere Application Server

► Installing the Tivoli Web Administration Tool

► Configuring the Tivoli Web Administration Tool

Installing Tivoli Web Administration Tool overview

Both the Tivoli Directory Server V5.2 and Tivoli Access Manager V5.1 include the Web Administration Tool used to manage the Tivoli Directory Server or Servers. The Web Administration Tool in both packages appears to be the same. There are, however, a couple of key differences between the two in the way that they are configured with packaged software components that we would like to point out:

▶ Web Administration Tool packaged with Tivoli Directory Server V5.2

The Tivoli Directory Server V5.2 Web Administration Tool is bundled with WebSphere Application Server Express V5.0.2, which does not provide the ability to enable WebSphere security. The Tivoli Directory Server V5.2 Web Administration Tool can be installed as part of the Tivoli Directory Server installation. Note that it is possible to install the Web Administration Tool on WebSphere Application Server V5.0.2 (not included with Tivoli Directory Server V5.2).

▶ Web Administration Tool packaged with Tivoli Access Manager V5.1

The Web Administration Tool packaged with Tivoli Access Manager V5.1 is bundled with WebSphere Application Server V5.0.2. When the Web Administration Tool is installed in the WebSphere Application Server, it installs the Web Administration Tool WAR file (IDSWebApp.war). After the installation of the WAR files, they must be deployed to the WebSphere Application Server application server (for example, server1) manually. The Web Administration Tool is used to manage the Tivoli Directory Server, and the Web Portal Manager is used to manage the Tivoli Access Manager (as an alternative, you can use the `pdadmin` command line interface).

For our example, we chose to install the Web Administration Tool provided with Tivoli Access Manager V5.1 on WebSphere Application Server V5.0.2, which will allow us to later configure WebSphere Application Server security.

The high-level steps to install and configure the Web Administration Tool are as follows:

▶ Installing WebSphere Application Server
▶ Installing the Tivoli Web Administration Tool
▶ Configuring the Tivoli Web Administration Tool

Installing WebSphere Application Server

This section describes how to install WebSphere Application Server V5.0 and Fix Pack 2 (V5.0.2) on the policy server node as a prerequisite to the Tivoli Access Manager V5.1 Web Administration Tool.

Installing WebSphere Application Server V5.0

To install the WebSphere Application Server V5.0, complete the following steps:

1. Insert the *Tivoli Access Manager V5.1 - Web Administration Tool* CD.

2. Navigate to the <CD_Root>\windows\websphere\nt directory and run `Install.exe` to start the WebSphere Application Server installer.

3. When the Select the desired language to be used for the installation wizard opens, select the desired language (for example, English) and click **OK**.

4. When the Welcome window opens, click **Next**.

5. When the License Agreement window opens, review the terms and, if in agreement, select **I accepts the terms in the license agreement**, and then click **Next**.

6. When the Setup Type window opens, select **Custom** and click **Next**.

7. When the Features Selection window opens, we selected the features displayed in Figure A-1 on page 284 and then clicked **Next**.
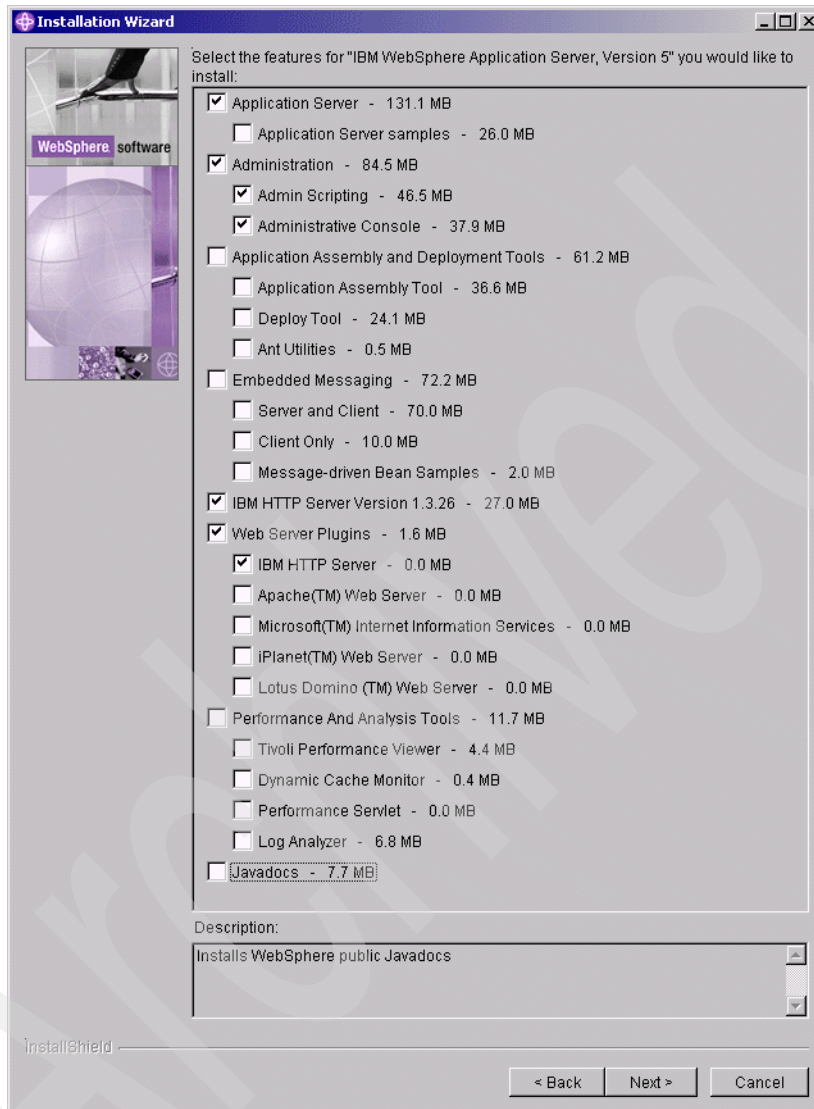
Figure A-1   WebSphere Application Server: Selected features

8. When the Features Installation directories window opens, we entered the
   following values and then clicked **Next**:

   – WebSphere Application Server: `c:\ibm\WebSphere\AppServer`
   – IBM HTTP Server: `c:\ibm\IBMHttpServer`

9. When the Node name and Host name window opens, we entered the following values and then clicked **Next**:

   – Node name: `phoenix`
   – Host name or IP Address: `phoenix.itsc.austin.ibm.com`

10. When the Windows Services window opens, we entered the following values and then clicked **Next**:

   – Clear Run WebSphere Application Server as a service option.
   – Select **Run IBM HTTP Server as a service**.
   – User ID: `administrator`
   – Password: `<password>`

11. When the Installation Summary window opens, review your selections and then click **Next** to begin copying files.

12. When the Register window opens, take the appropriate action.

13. When the First Steps window opens, click **Exit**.

14. Click **Finish** on the Installation Wizard page.

Installing WebSphere Application Server V5 Fix Pack 2 (V5.0.2)

The IBM WebSphere Application Server V5 Fix Pack 2 (V5.0.2) is a prerequisite to the Tivoli Directory Server V5.2 Web Administration Tool included with Tivoli Access Manager V5.1. WebSphere Application Server Fix Pack 2 is included on the *Tivoli Access Manager V5.1 - WebSphere Fixpack* CD or can be downloaded from the IBM WebSphere Application Server support site.

To install IBM WebSphere Application Server V5 Fix Pack 2, complete the following steps:

1. Ensure that you have stopped the WebSphere Application Server and all application servers and nodes.

2. Ensure that you have stopped the IBM HTTP Server and IBM HTTP Administration Windows service (WebSphere plug-in fixes).

   Note: The fix pack will attempt to update the IBM HTTP Server and will not be able to update the server if it is started.

3. Insert the *Tivoli Access Manager V5.1 - WebSphere Fixpack* CD.

4. Open a command window and navigate to the <CD_Root>\windows\websphere_fixpack directory.

5. Copy WebSphere Application Server V5 Fix Pack 2 to a temporary directory on the target system (for example, c:\temp\was5.fp2). Note that the WebSphere Update Installer Wizard needs write access.

6. Start the WebSphere Installation Update Wizard by running updateWizard.bat, found in the temporary directory (for example, c:\temp\was5.fp2).

7. When the WebSphere Update Installer language window opens, select the appropriate language for the wizard (for example, English) and then click **OK**.

8. When the Welcome window opens, click **Next**.

9. The WebSphere Update Installer should detect your current WebSphere Application Server version and installation directory (for example, c:\ibm\WebSphere\AppServer). Click **Next**.

10. Select **Install fix packs** and then click **Next**.

11. Enter the directory where you copied the fix pack. For example, we entered c:\temp\was5.fp2\fixpacks in the Fix pack directory text field, and then clicked **Next**.

12. Select the **was50_fp2_win** fix pack (default) and then click **Next**.

13. You will be prompted for the directories for the IBM HTTP Server and the WebSphere Application Server Embedded Messaging (not installed). We entered the following values and then clicked **Next**:

    – Select **IBM HTTP Server**.
    – IBM HTTP Server installation directory: c:\ibm\IBMHttpServer
    – Clear the Embedded Messaging options (not installed).

14. Review the fix pack settings and then click **Next** to begin the fix pack installation of files.

15. When the WebSphere Application Server V5 Fix Pack 2 installation is complete, click **Finish**.

Verifying WebSphere Application Server V5.0.2

To verify the functionality of the WebSphere Application Server V5.0.2 after the installation, complete the following steps:

1. Verify the WebSphere Application Server installation by selecting **Start** → **Programs** → **IBM WebSphere** → **First Steps**.

2. From First Steps window, click **Verify Installation**.

   If the server is not started, it will start the server and perform some tests. You will see console output with the status of each test run as passed.

> **Note:** Alternatively, you can start the Install Verification by opening a command prompt, navigating to the <was_home>\bin directory, and running `ivt.bat`.

3. After completing the Verify Installation, click **Exit** from the First Steps window.

4. Ensure that the IBM HTTP Server (WebSphere plug-in) is started.

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

```
cd \ibm\WebSphere\AppServer\bin
startServer 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 c:\ibm\WebSphere\AppServer\logs\server1\startServer.logs
> ```
>
> You should see the following message:
>
> ```
> Server server1 open for e-business
> ```
>
> The server1 directory will not get created until the first time the application server is started.

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

```
http://<was_hostname>:9090/admin
```

7. Log on to the WebSphere Administration Console (for example, admin).

8. When done verifying the WebSphere Administration Console, click **Logout** to close the Web browser.

9. Stop the server1 application server as follows:

```
cd \ibm\WebSphere\AppServer\bin
stopServer.bat server1
```

Installing the Tivoli Web Administration Tool

The installation of the Tivoli Directory Server Web Administration Tool packaged with Tivoli Access Manager V5.1 (*IBM Tivoli Access Manager Web Administration Interfaces for Windows 2000* CD) first installs the WAR file for the Web Administration Tool. After the WAR file is installed, the WAR file must be manually deployed to IBM WebSphere Application Server V5.0.2.

Installing Web Administration Tool

To install the Tivoli Web Administration Tool packaged with Tivoli Access Manager V5.1 on the policy server node, complete the following steps:

1. Insert the *IBM Tivoli Access Manager Web Administration Interfaces for Windows 2000* CD.

2. Navigate to the <CD_Root>\windows\Directory folder and run `Setup.exe` to start the installation.

3. When the Install Shield language window opens, select the appropriate language (for example, English), and then click **OK**.

4. When the Installing Current Version window opens, as shown in Figure A-2, click **No**.

Figure A-2   Web Administration Tool installation: Current version

5. When the Welcome window opens, click **Next**.

6. When Software License Agreement page opens, review the agreement and, if in agreement select **I accept the terms in the license agreement**. Click **Next**.

7. The installer will detect if application components have been installed. Review the information and click **Next**.

8. Select the language for the Tivoli Directory Server (for example, English) and click **Next**.

9. When the Select Features to Install window opens, we selected the following options and then clicked **Next**.

> **Note:** Only select **Web Administration Tool**.

   – Clear the Client SDK 5.2 option.
   – Select the **Web Administration Tool 5.2** option.
   – Clear the Server 5.2 option.
   – Clear the IBM WebSphere Application Server - Express 5.0.2 option.
   – Clear the DB2 V8.1 option.
   – Clear the GSKit option.

10. When the Summary window opens, click **Next** to begin the installation.

11. When the *Review readme* file opens, click **Next**.

12. When the installation is complete, you will be prompted to restart the system. This is not necessary, because we just installed the Web Administration Tool. Select **No, I will start my computer at a later time**. Note that it is not necessary in this case to restart your computer, because the installer only copied the Web Administration Tool War file (IDSWebApp.war). Click **Next**.

13. Click **Finish**.

Deploying Web Administration Tool on WebSphere Application Server

To deploy the Web Administration Tool on the WebSphere Application Server server1, complete the following steps:

1. Ensure that the WebSphere Application Server server1 is started. If not, start the server1 as follows:

   ```
   c:\ibm\WebSphere\AppServer\bin\startServer server1
   ```

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

   ```
   http://<hostname>:9090/admin
   ```

3. Log on to the Administration Console (for example, admin).

4. Select **Applications** → **Install New Application** in the console navigation tree.

5. When the Preparing for application installation window opens, select the following options and then click **Next**:

   – Path: **Local path**

   – Local Path: `c:\ibm\ldap\idstools\IDSWebApp.war`

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

   > **Note:** The file can either be on the client machine (the machine that runs the Web browser) or on the server machine (the machine to which the client is connected).

   – Context Root: `/IDSWebApp`

6. When the Generate bindings window opens, we accepted the default settings and clicked **Next**.

7. When the Step 1: Provide options to perform the installation window opens, we accepted the following default setting and clicked **Next**:

   – Application Name: `IDSWebApp_war` (default)

8. When the Step 2: Map virtual hosts for Web modules window opens, we selected the following options and then clicked **Next**:

   – Virtual Host: **default_host** (default)
   – Web Module: **IBM Tivoli Directory Server Web Application v2.0**

9. When the Step 3: Map modules to application servers window opens, we accepted the default and then clicked **Next**.

10. When the Step 4: Summary Review installation options window opens, click **Finish**.

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

12. When the Save to Master Configuration window opens, click **Save**.

13. Start the Tivoli Directory Server Web Administration Tool by selecting **Applications** → **Enterprise Applications**.

14. From the Enterprise Application window, select **IDSWebApp_war** and then click **Start**.

15. Click **Logout**.

Configuring the Tivoli Web Administration Tool

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

Defining the directory server node to the Web Administration Tool

To define the directory server with the Web Administration Tool, complete the following steps:

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

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

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

3. From the Web Administration Tool, enter the following values and then click **Login**:

   – LDAP Hostname: **Console Admin**
   – Username: `superadmin` (default)
   – Password: `secret` (default)

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 window opens, enter the following values and click **OK**:

     • Console administrator login: `webadmin`
      We created the administrator webadmin, but this could be any name you desire.
     • Current password: `<password>`
      (The default is secret.)

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

5. Add the Directory Server node:

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

   b. Click **Add**.

   c. Enter the Directory Server host name and change the port numbers if not using defaults, and click **OK**. We entered the following values:

     • Hostname: `phoenix.itsc.austin.ibm.com`
     • Port: `389`
     • Administration port: `3538`
     • Clear the SSL enabled option (default).

    You should see the new server listed after adding it.

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

Verifying the administration of IBM 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 by connecting to the directory server as follows:

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://localhost:9080/IDSWebApp/IDSjsp/Login.jsp`

3. From the Web Administration Tool, do the following and then click **Login**:

   – Select the newly created server (for example, phoenix.itsc.austin.ibm.com) from the drop-down list on the Login page.
   – Username: `cn=root`
   – Password: `<password>`

4. To start the Tivoli Directory Server, select **Server administration** → **Start/stop/restart server**.

5. Click **Start** (do not select Start/restart in configuration only mode).

   You should see the status message `Server started`.

Changing the password encryption method

After the installation, we recommend that you change the password encryption method from the default imask to SHA or crypt from the Web Administration Tool, as defined in *IBM Tivoli Directory Server Administration Guide V5.2*, SC32-1339. The primary reason is that imask is a two-way encoding format, and both SHA and crypt are one way.

We configured the password encryption as follows:

1. From the Web Administration Tool, select **Server administration** → **Manage security properties**.

2. From the Manage security properties window, click **Password policy**.

3. Select **SHA** from the Password encryption drop-down list and then click **OK** at the bottom of the page.

Abbreviations and acronyms

ACL	Access Control List	**JCE**	Java Cryptography Extension
AIX	Advanced Interactive Executive	**JIT**	Just In Time (compiler)
API	Application Programming Interface	**JKS**	Java Key Store
		JRE	Java Runtime Environment
ASCII	American Standard Codes for Information Interchange	**JSP**	Java ServerPage
		JVM	Java Virtual Machine
CMS	Common Management System	**KYR**	Keyring
CRM	Customer Resource Management	**LDAP**	Lightweight Directory Access Protocol
DES	Data Encryption Standard	**LDIF**	Lightweight Directory Interchange Format
DIIOP	Distributed IIOP	**LTPA**	Lightweight Third Party Authentication
DMZ	Demilitarized Zone		
DNS	Domain Name Service	**MD5**	Message Digest 5
DSAPI	Domino Security API	**MIME**	Multipurpose Internet Mail Extensions
EJB	Enterprise JavaBean		
ERP	Enterprise Resource Planning	**OID**	Object Identifier
ESE	Enterprise Server Edition	**PDJRTE**	Policy Directory Java Runtime Environment
FTP	File Transfer Protocol	**PKCS**	Public Key Cryptography Standards
HTML	Hypertext Markup Language		
HTTP	Hypertext Transfer Protocol	**RFC**	Request for Comments
HTTPS	HTTP Secure	**RPC**	Remote Procedure Call
IBM	International Business Machines Corporation	**SASL**	Simple Authentication and Security Layer
ITSO	International Technical Support Organization	**SDK**	Software Development Kit
		SOAP	Simple Object Access Protocol
J2EE	Java 2 Platform, Enterprise Edition		
		SSL	Secured Sockets Layer
J2SE	Java 2 Platform, Standard Edition	**SSO**	Single Sign-On
		TAI	Trust Association Interceptor
JAAS	Java Authentication and Authorization Services	**TCP/IP**	Transmission Control Protocol/Internet Protocol
JACL	Java Command Language	**UDB**	Universal Database

URL	Universal Resource Locator
VPN	Virtual Private Network
XML	Extensible Markup Language

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 299. Note that some of the documents referenced here may be available in softcopy only.

► *A Portal Composite Pattern Using WebSphere Portal V5*, SG24-6087

► *A Secure Portal Extended With Single Sign-On,* REDP-3743

► *A Secure Portal Using WebSphere Portal V5 and Tivoli Access Manager V4.1*, SG24-6077

► *Develop and Deploy a Secure Portal Solution Using WebSphere Portal V5 and Tivoli Access Manager V5.1*, SG24-6325

► *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014

► *IBM WebSphere Application Server V5.1 System Management and Configuration: WebSphere Handbook Series*, SG24-6195

► *IBM WebSphere Portal for Multiplatforms V5 Handbook*, SG24-6098

► *IBM WebSphere Portal V5: A Guide for Portlet Application Development*, SG24-6076

► *IBM WebSphere V5.0 Security: WebSphere Handbook Series*, SG24-6573

► *IBM WebSphere V5.1 Performance, Scalability, and High Availability WebSphere Handbook Series*, SG24-6198

► *Lotus Security Handbook*, SG24-7017

► *Understanding LDAP - Design and Implementation*, SG24-4986

► *WebSphere Studio Application Developer Version 5 Programming Guide*, SG24-6957

295

Other publications

These publications are also relevant as further information sources:

▶ *IBM Tivoli Access Manager Base Administration Guide, V5.1*, SC32-1360

▶ *IBM Tivoli Access Manager Base Installation Guide, V5.1*, SC32-1362

▶ *IBM Tivoli Access Manager Error Message Reference, V5.1*, SC32-1353

▶ *IBM Tivoli Access Manager for e-business Command Reference, V5.1*, SC32-1354

▶ *IBM Tivoli Access Manager for e-business IBM WebSphere Application Server Integration Guide, V5.1*, SC32-1368

▶ *IBM Tivoli Access Manager for e-business Performance Tuning Guide, V5.1*, SC32-1351

▶ *IBM Tivoli Access Manager for e-business Problem Determination Guide, V5.1*, SC32-1352

▶ *IBM Tivoli Access Manager for e-business Web Security Developer Reference, V5.1*, SC32-1358

▶ *IBM Tivoli Access Manager for e-business Web Security Installation Guide, V5.1*, SC32-1361

▶ *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide, V5.1*, SC32-1359

▶ *IBM Tivoli Access Manager Secure Sockets Layer Introduction and iKeyman Users Guide, V5.1*, SC32-1363

▶ *IBM Tivoli Directory Server Administration Guide, V5.2*, SC32-1339

▶ *IBM Tivoli Directory Server Installation and Configuration Guide, V5.2*, SC32-1338

▶ *IBM Tivoli Directory Server Performance Tuning Guide, V5.2*, SC32-1342

Online resources

These Web sites and URLs are also relevant as further information sources:

▶ WebSphere Portal documentation library

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

▶ *IBM WebSphere Portal for Multiplatforms Version 5.0.2 Information Center*

http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/

► WebSphere Portal V5.0 Fix Pack 2

`http://www.ibm.com/support/docview.wss?uid=swg24006309`

► WebSphere Application Server Cumulative Fix 2

`http://www.ibm.com/support/docview.wss?rs=203&context=SW000&uid=swg24005954`

► IBM HTTP Server Version 1.3.28.1 download page

`http://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2004-12-09+1`
`1%3A19%3A35.464785R&cat=&fam=&s=p&S_TACT=104CBW71&S_CMP=`

► Tivoli Access Manager for e-business Fix Pack 2

`http://www.ibm.com/support/docview.wss?rs=203&context=SSPREK&uid=swg24006925`

► WebSphere Portal Cumulative Fix 1

`http://www.ibm.com/support/docview.wss?rs=203&context=SSHRKX&uid=swg24006865`

► Java 2 Platform, Enterprise Edition - Documentation

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

► Lotus Documentation

`http://www.lotus.com/ldd/doc`

► Lotus Downloads

`http://www.lotus.com/ldd/down.nsf`

► Lotus Domino Documentation

`http://www.lotus.com/ldd/notesua.nsf/find/domino`

► Lotus Team Workplace Documentation

`http://www.lotus.com/ldd/notesua.nsf/find/quickplace`

► Lotus Instant Messaging and Web Conferencing Documentation

`http://www.lotus.com/ldd/notesua.nsf/find/sametime`

► DB2 UDB Version 8 Windows Fix Pack 4a and clients

`http://www.ibm.com/software/data/db2/udb/support/downloadv8W32fp4a.html`

► 32-bit DB2 for AIX 4.3.3 Fix Packs

`http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8fphist`

► "Configuring WebSphere Portal for DB2"

`http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_db2.html`

► "Setting up IBM Directory Server"

`http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_ids.html`

► "Configuring WebSphere Portal for IBM Directory Server"

`http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/cfg_ids_wp.html`

- ► "Configuring your Web server"

  `http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wpf/inst_ihs.html`

- ► "Collaborative portlets"

  `http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/wps/collabportlet.html`

- ► "Installing WebSphere Portal V5.0.2 on V5.0 platforms (Windows 2000 and UNIX)"

  `http://publib.boulder.ibm.com/pvc/wp/502/ent/en/readme/install_win_unix.html`

- ► "Installing WebSphere Portal Version 5.0.2 Cumulative Fix 1 (5.0.2.1)"

  `http://publib.boulder.ibm.com/pvc/wp/5021/ent/en/readme/install.html`

- ► *Sametime: How to Enable HTTP Tunnelling Over Port 80*, Technote 1090222

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21090222`

- ► *Troubleshooting Script for Setting QuickPlace to Search Across All Places*, Technote 1106449

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21106449`

- ► *Knowledge Collection: QuickPlace Issues Related to Sametime*, Technote 1115409

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21115409`

- ► *Troubleshooting Automatic Detection of your Mail File with the Different Lotus Collaborative Portlets*, Technote 1157029

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21157029`

- ► *Troubleshooting Pickers in Lotus Collaborative Portlets*, Technote 1157249

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21157249`

- ► *Troubleshooting WebSphere Portal, Sametime, QuickPlace and Domino SSO Issues*, Technote 1158269

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21158269`

- ► *Error: "Connection to QuickPlace Server Could not Be Established" in My Lotus Team Workplaces Portlet*, Technote 1159319

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21159319`

- ► *Troubleshooting Sametime Awareness in WebSphere Portal*, Technote 1163790

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21163790`

- ► *Password Errors When Using Web Conferencing Collaboration Center Portlet*, Technote 1170825

  `http://www.ibm.com/support/docview.wss?rs=0&uid=swg21170825`

- *Sun JVM Causes Browser to Hang on QuickPlace and Portal Server Pages*, Technote 1174295

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174295

- *Sporadic Error on Portal Pages with "Who Is Here" Portlet: "Failed to Contact Messaging Server"*, Technote 1174296

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174296

- *Lotus Collab Portlets: Sametime Contact List Portlet Sporadically Shows "Contact List Is Empty"*, Technote 1174300

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174300

- *Sametime Awareness in Lotus Collaborative Portlets Does not Function Consistently*, Technote 1174303

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174303

- *People Finder Portlet Does not Display 'Show Person Record' Menu Choice*, Technote 1174638

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174638

- *Sametime Meetings Created from Lotus Web Conf Portlet or QuickPlace Connect to Wrong Port*, Technote 1174639

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174639

- *"What's New" Link in 'My Lotus Team Workplaces' Portlet Shows Author's Full DN rather than Common Name*, Technote 1174643

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174643

- *"My Tasks" Link in Lotus Team Workplaces Portlet Displays "0" Instead of Correct Title*, Technote 1174645

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174645

- *QuickPlace Awareness Causes 'Who Is Here' Portlet To Show Names Multiple Times*, Technote 1174649

  http://www.ibm.com/support/docview.wss?rs=0&uid=swg21174649

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

Index

IBM

Redbooks

WebSphere Portal Collaboration Security Handbook

WebSphere Portal Collaboration Security Handbook

Describes the portal security environment

Integrates collaboration solutions

Scenario-based for identity solutions

Security is the hottest topic in the current Web-centric computing environment. This issue becomes the single largest concern for IT professionals who are stakeholders for Web applications, such as administrators, programmers, and users.

In this IBM Redbook, we discuss this security issue with the implementation of IBM WebSphere Portal Extend for Multiplatforms in an IBM Lotus collaborative environment. This discussion is scenario-based and aims to assist in the deployment of WebSphere Portal with Lotus Collaborative Components in a secure implementation. We describe several degrees of security, noting their advantages and disadvantages.

The primary goal of this scenario is to have a WebSphere Portal server with Lotus Team Workplace (formerly called QuickPlace) and Lotus Instant Messaging and Web Conferencing (formerly called Sametime) environment set up and running securely.

We discuss proxy authentication with IBM Tivoli Access Manager for e-business Version 5.1 and discuss the use of various identity providers, such as IBM Tivoli Directory Server, Domino LDAP, and Microsoft Active Directory.