



Sun Java™ System

# Instant Messaging Deployment Planning Guide

---

6 2004Q2

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Part No: 817-5937-10

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

# Contents

<b>About This Guide</b> .....	<b>5</b>
Who Should Read This Guide .....	5
What You Need to Know .....	6
How This Guide is Organized .....	6
Document Conventions .....	7
Monospaced Font .....	7
Bold Monospaced Font .....	7
Italicized Font .....	7
Square or Straight Brackets .....	8
Command-Line Prompts .....	8
Where to Find Related Information .....	8
Where to Find This Guide Online .....	9
References to Third-Party Websites .....	9
<b>Chapter 1 Overview</b> .....	<b>11</b>
What is an Instant Messaging Service? .....	12
Sun ONE Instant Messaging Core Product Components .....	12
Components Related to Instant Messaging .....	14
Web server .....	14
LDAP server .....	15
SMTP server .....	15
Sun Java System Calendar Server .....	15
Sun ONE Identity Server and Sun Java System Identity Server SDK .....	15
Sun Java System Portal Server .....	16
Supported Standards .....	17
Instant Messaging Architecture .....	19
The Instant Messaging Server .....	20
The Multiplexor .....	21
The Instant Messenger Client .....	22
Designing Your Deployment .....	23

Planning Process .....	23
Piloting Instant Messaging .....	25
Putting Instant Messaging Into Production .....	25
<b>Chapter 2 Deployment Examples .....</b>	<b>27</b>
Basic Instant Messaging Deployment .....	28
Authentication in a Basic Deployment .....	29
Instant Messaging With Email Notification .....	31
Instant Messaging with Calendar Alerts .....	33
Instant Messaging With Identity-based Server Policy Management or Single Sign On .....	35
Authentication in a Deployment With Identity Server Only .....	37
Instant Messaging With Portal-based Secure Mode or Archiving and Identity Server .....	39
Authentication in a Deployment With Portal Server .....	41
Instant Messaging With All Features Enabled .....	44
Physical Deployment Examples .....	44
Instant Messaging Server and Web Server on Separate Hosts .....	45
Multiple Multiplexors on Separate Hosts .....	46
Federation of Multiple Instant Messaging Deployments .....	47
<b>Chapter 3 Planning Your Sizing Strategy .....</b>	<b>49</b>
Collecting Sizing Data .....	50
Determine Peak Volume of Unique Logins .....	50
Create Your Usage Profile .....	50
Define Your User Base or Site Profile .....	54
Using a Load Simulator .....	55
System Performance Guidelines .....	55
Memory Utilization .....	56
Disk Throughput .....	56
Disk Capacity .....	57
Network Throughput .....	57
CPU Resources .....	58
Multiplexor Configuration Best Practices .....	58
Developing Architectural Strategies .....	59
Two-tiered Architecture .....	59
One-tiered Architecture .....	61
Example Resource Requirements .....	62
Small Deployment Sample Resource Requirements Numbers .....	62
Large Deployment Sample Resource Requirements Numbers .....	62
<b>Index .....</b>	<b>63</b>

# About This Guide

This guide explains how to deploy Instant Messaging and its accompanying software components.

Topics covered in this chapter include:

- [Who Should Read This Guide](#)
- [What You Need to Know](#)
- [How This Guide is Organized](#)
- [Document Conventions](#)
- [Where to Find Related Information](#)
- [Where to Find This Guide Online](#)
- [References to Third-Party Websites](#)

## Who Should Read This Guide

This guide is for individuals who are responsible for assessing and deploying Sun Java™ System Instant Messaging at your site, including:

- Planners
- Designers
- Architects
- Business Managers
- System Engineers

# What You Need to Know

This guide provides a framework for deployment best practices. It is assumed that you have a general understanding of the following:

- The Internet and the World Wide Web
- Web servers and clients
- LDAP servers such as the Sun Java System Directory Server
- Instant messaging and instant messaging concepts
- Basic administrative procedures of the operating system on which you are installing the software.
- Any other Java Enterprise System products such as Sun Java System Identity Server, Sun Java System Portal Server, and Sun Java System Messaging Server that you are planning to use with Instant Messaging.

## How This Guide is Organized

This guide contains the following chapters:

- About This Guide (this chapter)
- [Chapter 1, “Overview”](#)

This chapter provides an overview of basic Instant Messaging concepts and an overview of the Sun Java™ System Instant Messaging product. In addition, this chapter outlines a recommended deployment process of best practices: the concepts you should consider prior to deploying Instant Messaging.

- [Chapter 2, “Deployment Examples”](#)

Contains Instant Messaging deployment examples and describes how Instant Messaging interacts with other servers in your deployment.

- [Chapter 3, “Planning Your Sizing Strategy”](#)

This chapter provides a basic foundation for sizing your Instant Messaging deployment.

# Document Conventions

## Monospaced Font

Monospaced font is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

## Bold Monospaced Font

**bold monospaced font** is used to represent text within a code example that you should type. For example, you might see something like this:

```
./setup
```

In this example, **./setup** is what you would type from the command line.

## Italicized Font

*Italicized font* is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths and names and account IDs.

In the following URL, *webserver* and *webserverport* are variables:

```
http://webserver:webserverport
```

When you enter your information, substitute your web server name and web server port where the words *webserver* and *webserverport* appear italicized. For example, if your web server name is *i-zed*, and the web server port is *9980*, then the URL you would enter is:

```
http://i-zed:9980
```

## Square or Straight Brackets

Square (or straight) brackets [ ] are used to enclose optional parameters. For example, in this document you will see the usage for the `setup` command described as follows:

```
./setup [options] [argument]
```

It is possible to run the `setup` command by itself as follows to start the Instant Messaging installation:

```
./setup
```

However, the presence of *[options]* and *[arguments]* indicates that there are additional optional parameters that may be added to the `setup` command. For example, you could use `setup` command with the `-nodisplay` option to do an installation from the command line:

```
./setup -nodisplay
```

## Command-Line Prompts

Command-line prompts (for example, `%` for a C-Shell, or `$` for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command-line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

## Where to Find Related Information

In addition to this guide, Sun Java™ System Instant Messaging comes with the following supplementary information:

- Sun Java System Instant Messaging Release Notes
- *Sun Java System Instant Messaging Installation Guide*
- *Sun Java System Instant Messaging Administration Guide*

Sun Java™ System Instant Messaging also works with other products such as Sun Java System Directory Server. Documentation for this and other products is available online at the following URL:

<http://docs.sun.com/db/prod/sunone>



## Where to Find This Guide Online

You can find the *Sun Java System Instant Messaging Deployment Planning Guide* online in PDF and HTML formats at the following URL:

<http://docs.sun.com/db/prod/>

## References to Third-Party Websites

Sun is not responsible for the availability of third-party websites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

## References to Third-Party Websites

# Overview

Sun Java™ System Instant Messaging enables secure, real-time communication and collaboration, combining presence awareness with instant messaging capabilities such as chat, conferences, alerts, news, polls, and file transfers to create a rich collaborative environment. These features enable one-to-one as well as group collaboration through either short-lived communications or persistent venues such as conference rooms or news channels.

Instant Messaging ensures the integrity of communications through its multiple authentication mechanisms and secure SSL connections. Integration with the Sun Java System Portal Server and Sun Java System Identity Server bring additional security features, services-based provisioning access policy, user management, and secure remote access.

This chapter describes basic concepts you should understand prior to deploying Instant Messaging. It includes the following sections:

- [What is an Instant Messaging Service?](#)
- [Sun ONE Instant Messaging Core Product Components](#)
- [Components Related to Instant Messaging](#)
- [Supported Standards](#)
- [Designing Your Deployment](#)

# What is an Instant Messaging Service?

An instant messaging service is an open standards-based client-server solution that meets the instant messaging needs of enterprises and hosts of all sizes. It provides superior administration, scalability, performance, security, and connectivity throughout the enterprise and across the Internet.

At a simplistic level, an instant messaging service:

- Accepts instant messages from external sites.
- Determines the user to which the message should be delivered, and routes it accordingly.
- Accepts instant messages from internal hosts.
- Determines the destination system to which the message should be delivered, and routes it accordingly.

In addition, an instant messaging service can provide real-time conferencing, news and calendar alerts, and for offline users, email message forwarding.

Of crucial importance to a good instant messaging service is that the service follows the SHARP (scalability, high availability, reliability, and good performance) standard.

## Sun ONE Instant Messaging Core Product Components

Sun Java™ System Instant Messaging contains the following core components:

- *Instant Messenger resources (client)*, a set of files that make up the client program for end users to initiate, compose, and reply to messages. Typically users also use the client to participate in conferences. The client is also called Sun Java™ System Instant Messenger.
- *Instant Messaging server*, an electronic message delivery system that supports instant message delivery from one system to another system. The server serves the presence information to Instant Messenger clients, allows end users to establish sessions, and enforces policies.
- *Instant Messaging multiplexor*, a scalability component that consolidates messenger connections. In order to support large deployments, for example 40,000 concurrent connections, Instant Messaging uses a connection multiplexor to improve server scalability. This component opens a single

connection to the Instant Messaging server. In addition to scalability, you can install the multiplexor outside the firewall while leaving the server inside the firewall in order to protect from unauthorized external access. The Instant Messaging multiplexor is also referred to as the multiplexor.

- *Access, Communication, and Transfer Protocols*, such as LDAP, HTTP, TCP/IP, and SMTP protocols. These protocols can be found in [Supported Standards](#).
- *Sun Java™ System Identity Server Instant Messaging Service Definition*, a service used by the Sun Java System Identity Server and the Sun Java System Identity Server SDK to provide policy management and single sign-on capabilities for Instant Messaging.
- *Instant Messaging API*, allows you to create custom Instant Messaging clients.
- *LDAP server*, a “lightweight” version of the Directory Access Protocol (DAP). While typically not an instant messaging server component, the LDAP server is integral for user maintenance and management. It usually serves the needs of a wide variety of applications in an organization.

---

**NOTE** Since a proper Directory Server implementation is instrumental to a successful Instant Messaging deployment, it is highly recommended that you read the *Sun Java System Directory Server Deployment Guide* in addition to this guide.

---

# Components Related to Instant Messaging

The software components discussed in this section work with Instant Messaging server, but are installed separately. [Chapter 2, “Deployment Examples”](#) provides more detailed information that illustrates how these servers interact with Instant Messaging.

## Web server

(Required) For a basic deployment, you need to install a web server, such as Sun Java™ System Application Server SE (Standard Edition). Deployments with Sun Java System Portal Server use the web server that ships with Portal Server. In either case, the Instant Messenger resources must reside on the web server host.

Instant Messaging requires a web server to serve the Instant Messenger resources. The Instant Messenger resource files include:

- The `index.html` file, provided by Instant Messenger, or a home page with a link to invoke Sun Java™ System Instant Messenger.
- Sun Java™ System Instant Messenger jar files (`messenger.jar`, `imres.jar`, `imbrand.jar`, `imdesktop.jar`, `imnet.jar`, and `imjni.jar`).
- The Sun Java™ System Instant Messenger Online Help.

You must install Instant Messenger resources on the same host where the web server is installed. In a Sun Java System Identity Server deployment, you can install these resources on the Identity server’s host or on a different web server host. In most cases, the resources will be installed on the same host where you installed the Instant Messaging server software. It is possible to locate the Instant Messenger resources on a host other than the Instant Messaging server or multiplexor. For more information on this, see *Sun Java System Instant Messaging Installation Guide*.

---

**NOTE**      Install the web server before installing Sun ONE Instant Messaging.

If you are using Sun Java System Portal Server, you can use the web server that is shipped with the product. You do not need to install a separate web server for Instant Messaging.

---

## LDAP server

(Required) Instant Messaging uses an LDAP server, such as Sun Java System Directory Server, for end user authentication and search. In a deployment with Sun Java System Portal Server, Instant Messaging uses the same LDAP server used by the Portal server.

The Instant Messaging server does not store the Instant Messenger end-user authentication information; instead, this information is stored in the LDAP server.

By default, the Sun Java™ System Instant Messaging server relies on the common end-user attributes `cn` and `uid` to search for end-user and group information. If you want, you can configure the server to use another attribute for search. In addition, Sun Java™ System Instant Messaging properties (such as contact lists and subscriptions) can be stored in files on the Sun Java™ System Instant Messaging server or in the LDAP server.

If you do not have an LDAP directory installed, you must install one. For more information, see *Sun Java System Instant Messaging Installation Guide*. For instructions on configuring the server to use a non-default attribute for user search, see the *Sun Java System Instant Messaging Administration Guide*.

## SMTP server

(Optional) A Messaging server, such as Sun Java System Messaging Server, is used to forward instant messages, in the form of email, to end users who are offline. The SMTP server is not shipped with Instant Messaging.

## Sun Java System Calendar Server

(Optional) The Sun Java System Calendar Server, is used to notify users of calendar-based events. The Calendar server is not shipped with Instant Messaging.

## Sun ONE Identity Server and Sun Java System Identity Server SDK

(Optional, Solaris only) Sun Java System Identity Server and Sun Java System Identity Server SDK provide end user and service management, authentication and single sign-on services. They also provide policy management, logging service, debug utilities, the admin console, and client support interfaces.

In addition, Sun Java System Identity Server and SDK are required in deployments that include Sun Java System Portal Server. In both deployments, the SDK must be installed on the Instant Messaging server's host.

## Sun Java™ System Portal Server

(Optional, Solaris only) Sun Java™ System Portal Server supports message archiving, and allows you to run Instant Messaging in secure mode. In addition, the Instant Messenger client is made available to end users through the Portal Server desktop. The following two components of Sun Java System Portal Server provide additional functionality:

- [Portal Server Desktop](#)
- [Sun ONE Portal Server, Secure Remote Access](#)

### Portal Server Desktop

Sun Java™ System Instant Messenger installed in the Portal Server environment can be launched from the Instant Messaging channel available to end users on Portal Server Desktop.

### Sun ONE Portal Server, Secure Remote Access

Sun ONE Portal Server, Secure Remote Access enables remote end users to securely access their organization's network and its services over the Internet for Solaris-based or Windows-based systems. The end user can access Secure Remote Access by logging into the web-based Portal Server Desktop through the portal gateway. The authentication module configured for Sun ONE Portal Server authenticates the end user. The end-user session is established with Sun ONE Portal Server and the access is enabled to the end user's Portal Server Desktop.

In the Sun Java System Portal Server environment, you can configure Instant Messenger in either secure or non-secure mode. In the secure mode, communication is encrypted through the Sun Java System Portal Server Netlet. When you are accessing Instant Messenger in the secured mode, a lock icon appears in the Status area of the Instant Messenger. In the non-secure mode, the Instant Messenger session is not encrypted. For more information on Netlet, see *Sun ONE Portal Server, Secure Remote Access Administrator's Guide*.



# Supported Standards

Instant Messaging is built on native Internet technology, so you can maintain a single architecture inside and outside your organization, even when collaborating with your customers and partners. Additionally, you aren't locked into a proprietary system. All key components of Instant Messaging are based on proven, open Internet standards such as:

- **LDAP.** Provides access to enterprise directory information, enabling an accurate, secure instant messaging system.
- **HTML.** Formatting language for providing web browser access for the client.
- **HTTP.** HyperText Transport Protocol for providing web browser access for the client.
- **SMTP.** Simple Mail Transfer Protocol for reliable delivery of instant messages over Internet mail messages.
- **TCP/IP.** Proven, worldwide networking protocol.

## Instant Message Structure Format

HTML (HyperText Markup Language) is typically used as a standard for web documents. The Instant Messenger client also formats instant messages using HTML. This allows users to include hyperlinks within messages.

## Access Protocol

In Instant Messaging, user information and preferences are retrieved from an LDAP directory. This directory can be dedicated for use by Instant Messaging, or the Sun Java System Portal Server's directory. User data is typically retrieved using LDAP search functions.

LDAP provides a common language that client applications and servers use to communicate with one another. LDAP is a "lightweight" version of the Directory Access Protocol (DAP) used by the ISO X.500 standard. DAP gives any application access to the directory via an extensible and robust information framework, but at an expensive administrative cost. DAP uses a communications layer that is not the Internet standard TCP/IP protocol and has complicated directory-naming conventions.

LDAP preserves the best features of DAP while reducing administrative costs. LDAP uses an open directory access protocol running over TCP/IP and uses simplified encoding methods. It retains the X.500 standard data model and can support millions of entries for a modest investment in hardware and network infrastructure.

## Communication and Message Transfer Protocols

Server-to-server and client-to-server communications occur over TCP/IP.

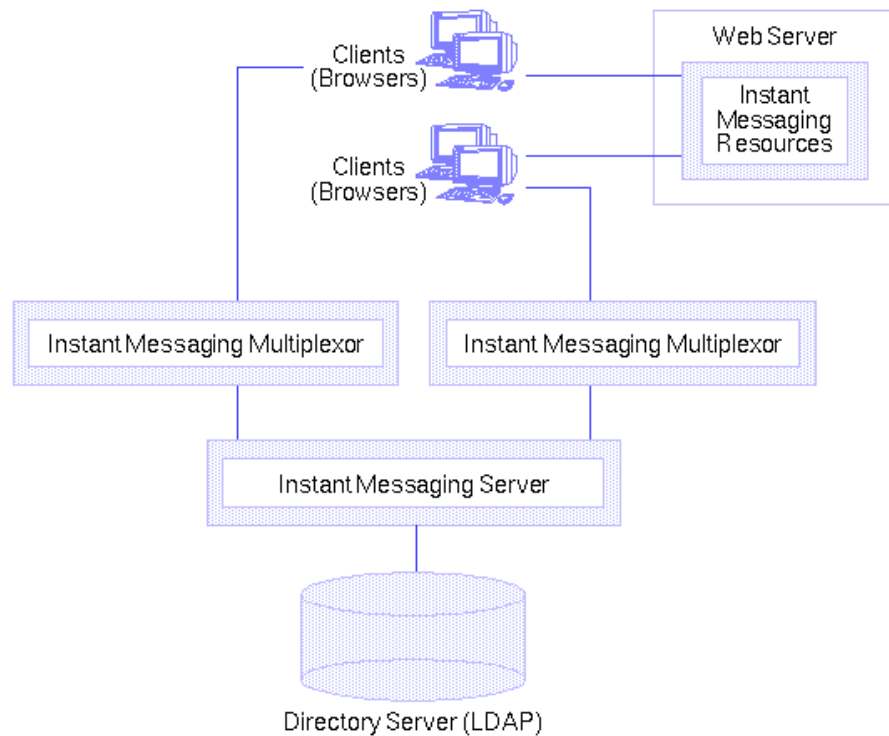
A message transfer protocol is used to send messages to offline users. SMTP is the most commonly used protocol.

Browsers use HTTP to retrieve Instant Messenger resource files from the Web server. Once retrieved, the browser reads the HTML and displays the contents of the files.

# Instant Messaging Architecture

Figure 1-1 shows the basic out-of-the-box Sun Java™ System Instant Messaging architecture.

Figure 1-1 Sun Java™ System Instant Messaging Basic Architecture



The Web server (or an application server with a Web service embedded), downloads the Instant Messaging resources via a browser to the clients. The resource files make up the client. Clients send messages to one another through a multiplexor which forwards the messages on to the Instant Messaging server.

The Directory server stores and retrieves local user and group delivery information such as preferences, location, and to which multiplexor to route messages for this user. When the Instant Messaging server receives a message, it uses this information to determine where and how the message should be delivered. In addition, the Directory server may contain user information such as contact lists and subscriptions.

In this basic configuration, Instant Messaging directly accesses a Directory Server to verify user login name and passwords for mail clients that use Instant Messaging.

Outgoing instant messages from clients go directly to the multiplexor. The multiplexor sends the message to the appropriate Instant Messaging server, which in turn forwards the message to another Instant Messaging server, or if the message is local, to the multiplexor with which the recipient is associated. (See [“Physical Deployment Examples” on page 44](#) for illustrations of this process.)

New users are created by adding user entries to the directory. Entries can be created or modified by modifying the directory using the tools provided with the Directory server.

Instant Messaging components are administered using a set of command line interfaces and text-based configuration files. Any machine connected to the Instant Messaging host can perform administrative tasks (assuming, of course, the administrator has the required privileges).

The following sections outline the three primary components of Instant Messaging in further detail:

- [The Instant Messaging Server](#)
- [The Multiplexor](#)
- [The Instant Messenger Client](#)

## The Instant Messaging Server

The Instant Messaging server handles tasks such as controlling Instant Messenger privileges and security, enabling Sun Java™ System Instant Messenger clients to communicate with each other by sending alerts, initiating chat conversations, and posting messages to the available news channels.

The Instant Messaging server supports the connection of a multiplexor that consolidates connections over one socket. For more information on the multiplexor, see [“The Multiplexor”](#).

Access control files and Sun Java System Identity Server policies are used for administration of end users, news channels, and conference rooms.

The Instant Messaging server routes, transfers, and delivers instant messages for the Sun Java™ System Instant Messaging product.

### Direct LDAP Lookup

The server can look up directory information directly from the LDAP server. The results of the LDAP queries are cached in the process, with configurable aging and expiration, so settings are tunable. Refer to the *Sun Java™ System Directory Server Administrator's Guide* for further information.

### Message Delivery

After the message is processed, the server sends the message to the next stop along the message's delivery path. This may be the intended recipient's multiplexor, or another server. Once received by a multiplexor, the message is routed directly to the intended recipient. (See [“Physical Deployment Examples” on page 44](#) for illustrations of this process.)

## The Multiplexor

The Instant Messaging multiplexor component connects multiple instant messenger connections into one TCP (Transmission Control Protocol) connection, which is then connected to the Instant Messaging server. The multiplexor reads data from Instant Messenger and writes it to the server. Conversely, when the server sends data to Instant Messenger, the multiplexor reads the data and writes it to the appropriate connection. The multiplexor does not perform any end user authentication or parse the client-server protocol (IM protocol). Each multiplexor is connected to one and only one Instant Messaging server.

You can install multiple multiplexors based on your deployment requirements. You must install at least one multiplexor. For more information, see [Chapter 2, “Deployment Examples”](#).

## The Instant Messenger Client

The Java-based Sun Java™ System Instant Messenger is Instant Messaging's client that can be configured to be a browser-based applet using Java Plug-in, or a standalone Java application using Java Web Start.

To run Instant Messenger client on Solaris, you must use Java Web Start. On Microsoft Windows you can run Instant Messenger as an applet or a Java Web Start application. It is recommended that you run Sun ONE Instant Messenger as a Java Web Start application.

For more information on customizing Sun Java™ System Instant Messenger, see the *Sun Java System Instant Messaging Administration Guide*.

Instant Messenger provides the following modes of communication:

- **Chat** - Sun Java™ System Instant Messenger's version of Instant Messaging conferences is called chat. Chat is a real-time conversation capability that enables end users to complete projects, answer customer queries, and complete other time-critical assignments. Chat sessions (two or more participants) are held in chat rooms created on a need basis.
- **Conference Rooms** - Conference rooms are persistent chat rooms that work similarly to regular chat sessions, but offer:
  - Access Control
  - Moderated Chats
- **Alerts** - Alerts enable information delivery and response to end users through the Instant Messenger interface. Alerts can deliver time-critical information to the end user. The sender of the alert message is notified when the message is delivered, and read by the recipient.
- **Poll** - The polling function enables you to ask end users for their response to a question. You can send a question and possible answers to poll recipients, and the recipients can respond with their selected answer.
- **News** - News channels are forums for posting and sharing information. End users can subscribe to news channels of interest to see updates using the URL of the news channels or view the news channel updates through static messages. Administrators control news channel access by assigning end users to the channels they need, and deciding who can see or post information to the channels.

---

**NOTE** The instant messages can contain embedded URLs. If you are using proxy servers, you may need to have clients using Java Web Start modify their proxy configuration for resolving such URLs.

For more information on configuring the proxy settings manually, see the *Sun Java System Instant Messaging Administration Guide*.

---

## Designing Your Deployment

During the planning process, you will gather data about your requirements, such as environment and data sources. With this information, you can design an Instant Messaging deployment that meets the needs of your users.

The flexibility of Sun Java™ System Instant Messaging allows you to rework your design to meet unexpected or changing requirements, even after deployment.

The following topics outline the stages in a successful deployment:

- [Planning Process](#)
- [Piloting Instant Messaging](#)
- [Putting Instant Messaging Into Production](#)

### Planning Process

The process for planning your deployment can be broken into the following functions:

- **Analyzing Your Requirements**  
This is one of the first steps in planning your deployment.
- **Determining Your Infrastructure Needs**  
For example network capacity, backup infrastructure, and DNS services.
- **Designing Your Topology**  
Topology design involves determining how you divide your system among multiple servers and how these servers communicate with one another.

- **Planning Your Directory Data**  
You need to understand the relationships between your directory tree, schema, and Instant Messaging.
- **Developing an Instant Messaging Architecture**
- **Designing Secure Instant Messaging**  
You need to plan how to protect Instant Messaging from common security threats.
- **Creating a Monitoring Strategy**  
Using SNMP and other methods to monitor your deployment. A well-planned deployment will perform without extensive intervention from an administrator.
- **Planning for Growth**  
As the number of users and messages increase, it is important to outline successful guidelines for capacity planning.
- **Identifying System Requirements**  
Often, people make the mistake of thinking about their system requirements before planning the rest of the deployment. In actuality, you should define your system requirements only after considering the aforementioned topics. In particular, software, hardware, and input/output sub-systems.
- **Planning Your Sizing Strategy**  
This chapter explains basic sizing concepts. [Chapter 3, “Planning Your Sizing Strategy”](#) outlines the factors to consider when sizing your deployment.



## Piloting Instant Messaging

After you have designed your deployment, you can start the deployment process. The first step of the deployment phase is installing a server instance as a pilot and testing whether Instant Messaging can:

1. Send and route messages
2. Handle the user load
3. Perform directory lookups for user information
4. Create and support conferences

If the deployment is not adequate as it is, you can adjust your pilot design until you have a robust service you can confidently introduce to your enterprise.

## Putting Instant Messaging Into Production

Once you have piloted and tuned Instant Messaging, you need to develop and execute a plan for taking Instant Messaging from a pilot to production. Create a production plan that includes the following:

- Estimate of the resources you need
- List of tasks you must perform before installing servers
- Schedule of what needs to be accomplished and when
- Set of criteria for measuring the success of your deployment

For information on installing Instant Messaging, refer to the *Sun Java System Instant Messaging Installation Guide*. For information on administering and maintaining Instant Messaging, refer to the *Sun Java System Instant Messaging Administration Guide*.



# Deployment Examples

This chapter contains descriptions of Instant Messaging deployment examples. Depending on the features you want to implement in your deployment, you will need to install different sets of servers. For example, to support email notification, you need to install an SMTP server, if you do not want to support email notification, you don't need the SMTP server. This section describes some deployment options based on feature sets. For information on supported software and versions, see the *Sun Java™ System Instant Messaging Release Notes*.

---

**NOTE** For this release, all deployment options are available on Solaris, and a subset of the options are available on Linux and Windows operating systems.

---

For more detailed information about servers that interoperate with Instant Messaging, see [“Components Related to Instant Messaging”](#) on page 14. The examples are discussed in the following sections:

- [Basic Instant Messaging Deployment](#)
- [Instant Messaging With Email Notification](#)
- [Instant Messaging with Calendar Alerts](#)
- [Instant Messaging With Identity-based Server Policy Management or Single Sign On](#)
- [Instant Messaging With Portal-based Secure Mode or Archiving and Identity Server](#)
- [Instant Messaging With All Features Enabled](#)

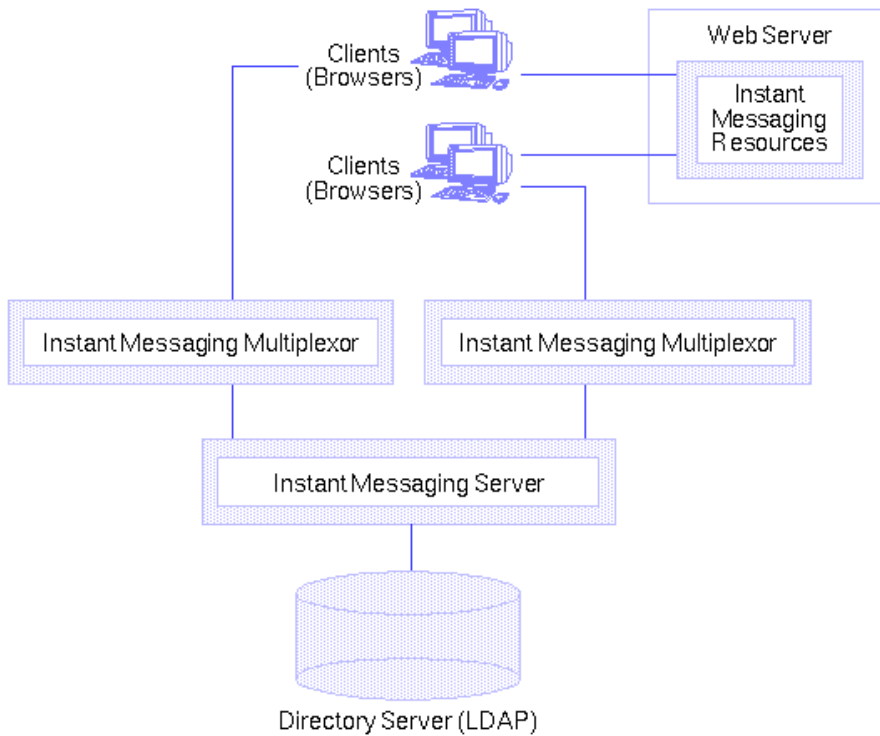
# Basic Instant Messaging Deployment

This option provides basic Instant Messaging functionality such as instant chat, news alerts, and conferences. To provide this basic functionality, you need to install the following:

- Instant Messaging server and one or more multiplexors
- Instant Messaging resources
- Web server such as Sun Java System Web Server
- LDAP Server such as Sun Java System Directory Server

Figure 2-1 shows what this basic deployment looks like.

**Figure 2-1** Basic Instant Messaging Deployment



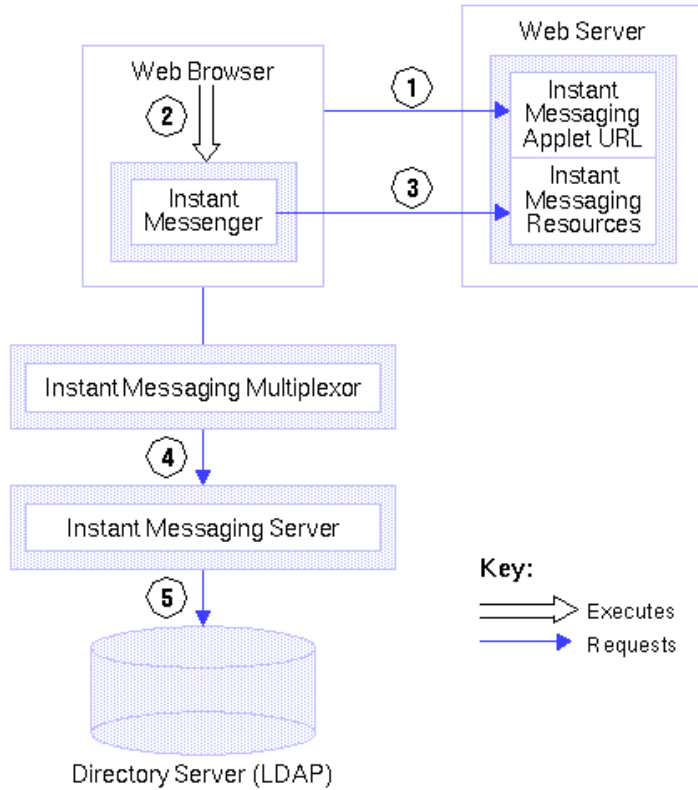
In this example:

- The LDAP server provides user entries for authentication and lookup.
- The clients download the Instant Messaging resources from a Web server (or Application server).
- Clients always connect to the Instant Messaging server through a multiplexor.

## Authentication in a Basic Deployment

[Figure 2-2 on page 30](#) illustrates the interaction of the software components in the authentication process of a basic deployment of Instant Messaging. The focus is on the flow of authentication requests, where the protocols used for requests are indicated above the arrows. The Instant Messaging protocol is a proprietary protocol. An explanation of the steps in this process follow the figure.

**Figure 2-2** Flow of Authentication Requests in a Basic Deployment



The authentication process in a basic deployment works as follows:

1. End user accesses the Sun Java™ System Instant Messenger applet URL from a browser and chooses a method to invoke the client.
2. The browser invokes Java Web Start or the Java Plug-in.
3. Java Web Start or the Java plug-in downloads the necessary Sun Java™ System Instant Messenger resource files and starts Instant Messenger.
4. The log-in window appears and the end user enters the log-in name and password. The log-in data is sent to the Instant Messaging server through the multiplexor.
5. The Instant Messaging server communicates with the LDAP server to authenticate the end user and to request end-user information, such as contact lists or subscriptions.

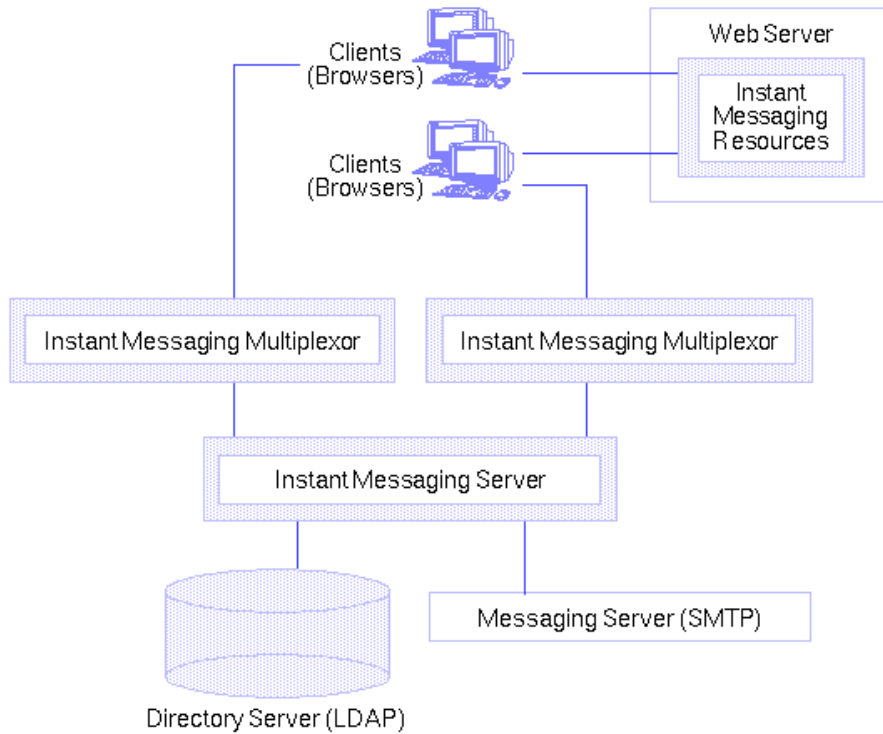
When the end-user authentication is complete, the Instant Messaging main window appears, displaying the contact list for the end user. The end user can now start and participate in Instant Messaging sessions with the other end users.

## Instant Messaging With Email Notification

This option provides the same functionality as [Basic Instant Messaging Deployment](#) and also supports email notification to offline users. To provide this functionality, you need to include the servers listed in [Basic Instant Messaging Deployment](#) and also install an SMTP server such as Sun Java System Messaging Server in your deployment. To enable this feature, you are prompted to identify the SMTP server to use with Instant Messaging during installation. If you do not have an SMTP server installed, you must install one before installing the Instant Messaging software. [Figure 2-3](#) shows Instant Messaging with email notification enabled on the network.

Authentication flow in this deployment is the same as in a basic deployment. For information on authentication flow, see “Authentication in a Basic Deployment” on page 29.

**Figure 2-3** Instant Messaging Deployment with Email Notification



In this example:

- The LDAP server provides user entries for authentication and lookup.
- The Instant Messaging server forwards messages intended for offline users to the SMTP server. The SMTP server then sends the message as an email to the user's mailbox.
- The clients download the Instant Messaging resources from a Web server (or Application server).
- Clients always connect to the Instant Messaging server through a multiplexor.



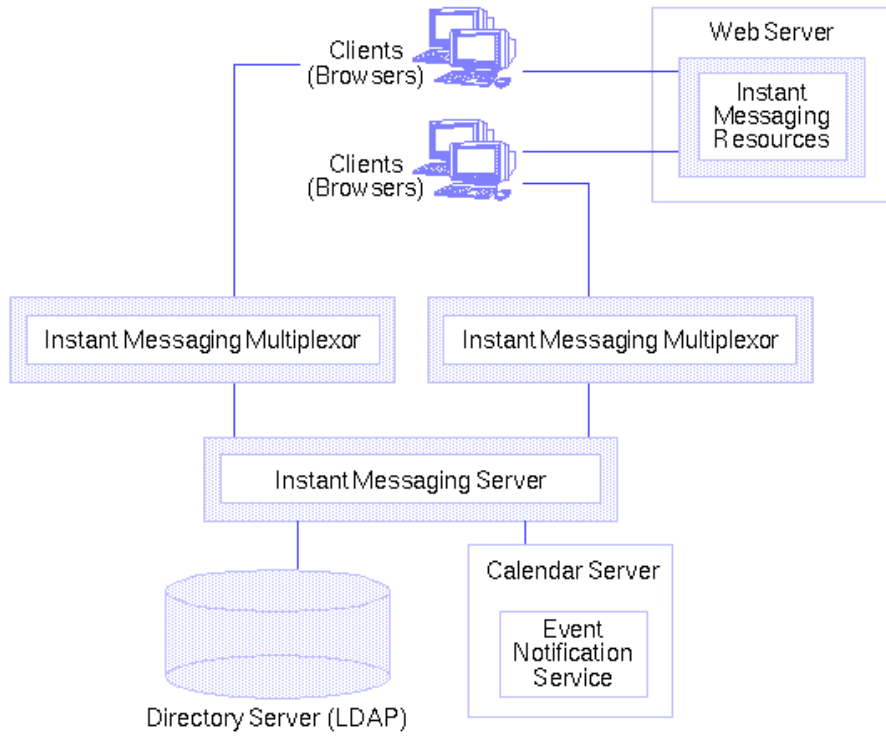
# Instant Messaging with Calendar Alerts

This option provides the same functionality as [Basic Instant Messaging Deployment](#) and also supports Instant Messaging based notification of calendar events to users. To provide this functionality, you need to include the servers listed in [Basic Instant Messaging Deployment](#) and also install Sun Java System Calendar Server in your deployment. To enable this feature, you are prompted to identify the Calendar server to use with Instant Messaging during installation. If you do not have a Calendar server installed, you must install one before installing the Instant Messaging software. [Figure 2-4](#) shows Instant Messaging with calendar notification enabled on the network.

See the Sun Java System Calendar Server documentation for more information about that product.

Authentication flow in this deployment is the same as in a basic deployment. For information on authentication flow, see “Authentication in a Basic Deployment” on page 29.

**Figure 2-4** Instant Messaging with Calendar Alerts



In this example:

- The LDAP server provides user entries for authentication and lookup.
- The Event Notification Server (ENS) sends notifications of calendar events to the Instant Messaging server which then forwards the notification on to the appropriate end user.
- The clients download the Instant Messaging resources from a Web server (or Application server).
- Clients always connect to the Instant Messaging server through a multiplexor.

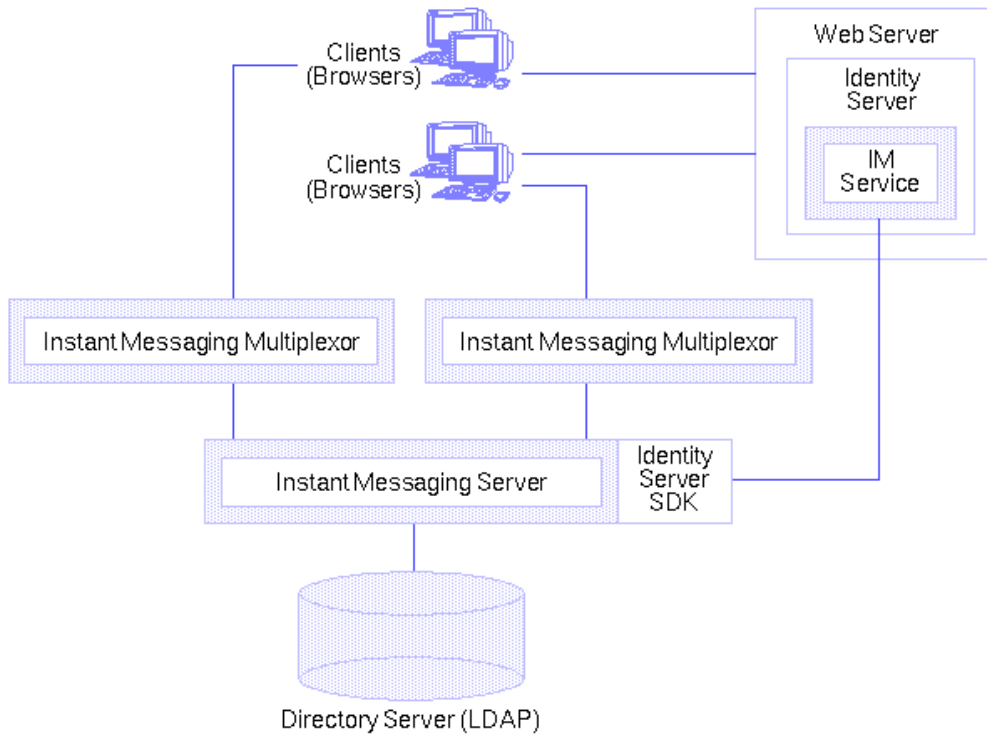
## Instant Messaging With Identity-based Server Policy Management or Single Sign On

(Solaris only) This option provides the same functionality as [Basic Instant Messaging Deployment](#) and also allows you to access and use Sun Java System Identity Server policy features and single sign on. To provide this functionality, you need to include the servers listed in [Basic Instant Messaging Deployment](#) and also install Sun Java System Identity Server in your deployment. In addition, you need to install Sun Java System Identity Server SDK on the Instant Messaging server's host.

In this case, Instant Messaging uses the directory to search for users but not to authenticate or authorize them. Instead, the Sun Java System Identity Server is responsible for authenticating and authorizing users.

[Figure 2-5](#) shows the Identity-based deployment option.

**Figure 2-5** Instant Messaging With Identity-based Server Policy Management or Single Sign On



In this example:

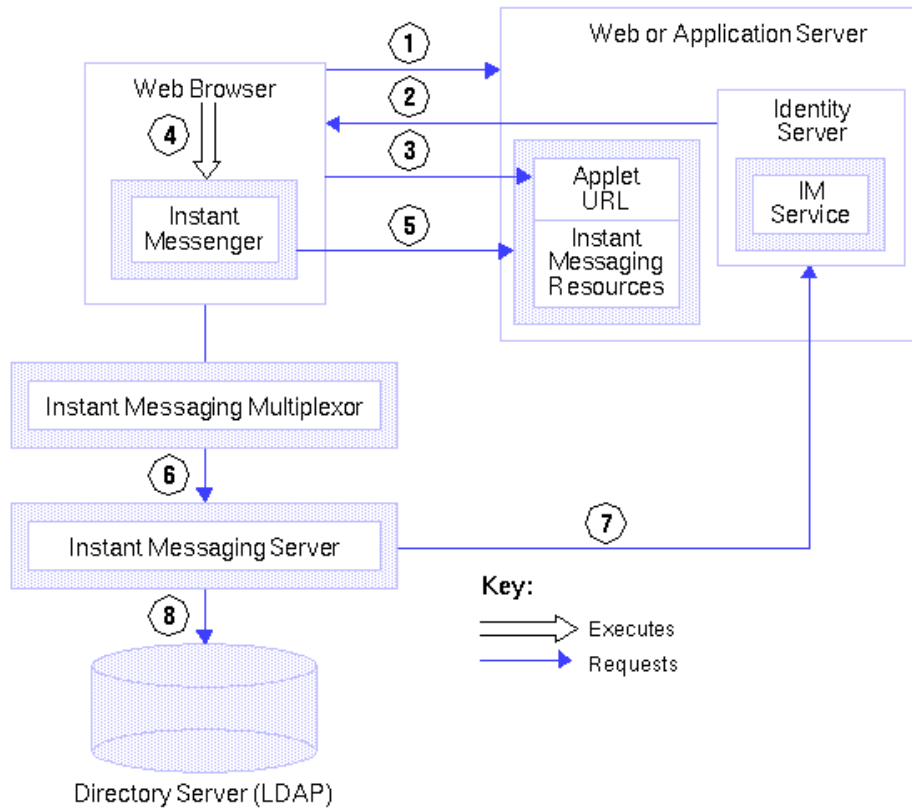
- LDAP Server provides user entries.
- The Web Server (can also be an Application Server, which includes a Web Server) downloads the Instant Messaging resources via a browser to the clients. The Resources are basically the client.
- Clients are always connect through a multiplexor.
- Instant Messaging related services provided by the Identity server include the presence service and the Instant Messaging service.

- The Web server can be used to access the Identity server administration interface which manages the identity based services for the Instant Messaging deployment. The web server for Identity server can be the same as the one that serves the Instant Messaging resources. See the Sun Java System Identity Server documentation for more details.
- Identity Server SDK provides the API used by the Instant Messaging server to interact with the Identity server.

## Authentication in a Deployment With Identity Server Only

[Figure 2-8](#) illustrates authentication process used by the Instant Messaging software in collaboration with the Sun Java System Portal Server and Sun Java System Identity Server components in a Single Sign-On environment. As with [Figure 2-2 on page 30](#), this figure focuses on the flow of authentication requests. An explanation of the steps in this process follows the figure.

**Figure 2-6** Flow of Authentication Requests in an Identity Server Configuration



The authentication process of the Instant Messaging server in this deployment within a single sign-on environment works as follows:

1. The end user logs in to the Sun Java System Identity Server by entering the URL in a web browser.
2. The Sun Java System Identity Server software authenticates the end user and returns a session token.

The session token is what enables single sign-on to work. This token is provided as an applet parameter and is used throughout the authentication process. End users are not asked for their credentials again as long as the session token is present.

3. End user accesses the Sun Java™ System Instant Messenger applet URL from a browser and chooses a method to invoke the client.
4. The browser invokes Java Web Start or the Java Plug-in as appropriate.
5. Java Web Start or the Java plug-in downloads the necessary Sun Java™ System Instant Messenger resource files and starts Instant Messenger.
6. Instant Messenger requests authentication to the Instant Messaging server using the session token.
7. The Instant Messaging server asks Sun Java System Identity Server to validate the session token. If the session is valid, Instant Messenger displays the end user's contact list and the end user can then use Instant Messenger services: chat, alerts, polls, etc.
8. The Instant Messaging server must query LDAP directly to get or set end-user information, such as contact lists or subscriptions.

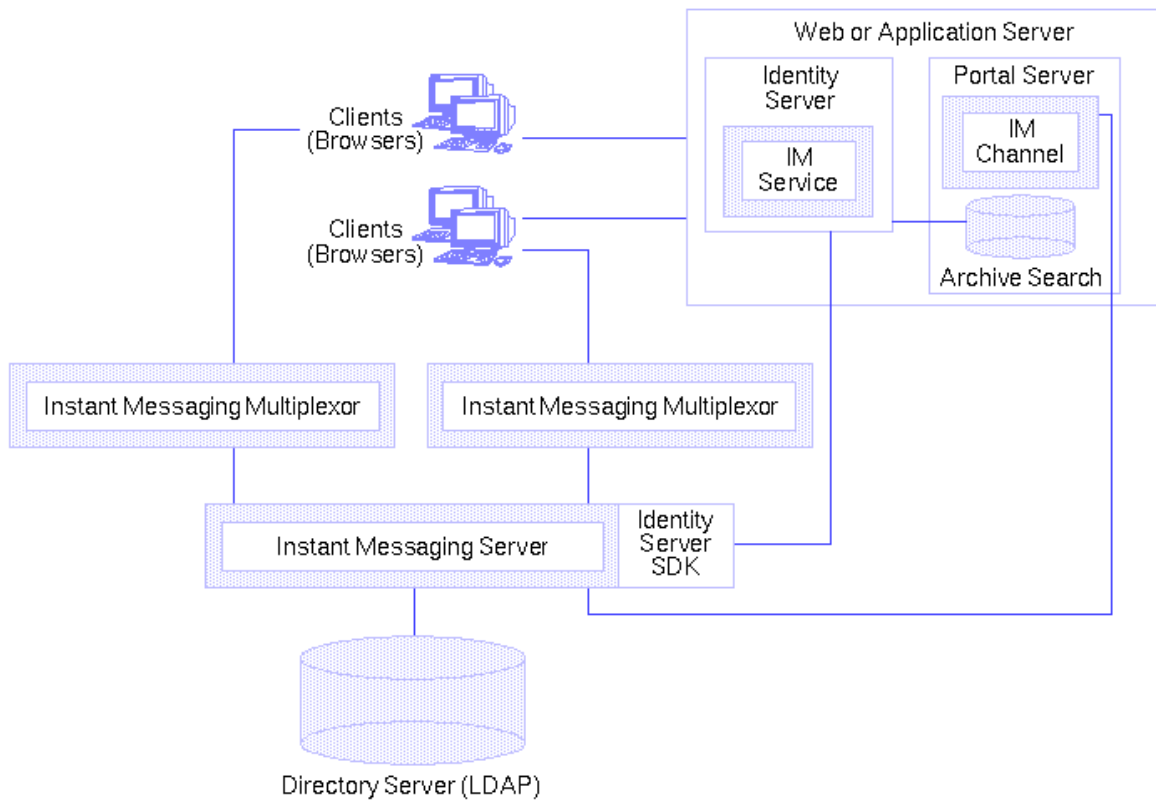
## Instant Messaging With Portal-based Secure Mode or Archiving and Identity Server

(Solaris only) This option provides the same functionality as [Basic Instant Messaging Deployment](#) and also supports message archiving, and allows you to run Instant Messaging in secure mode. In addition, the Instant Messenger client is made available to end users through the Portal Server desktop. To provide this functionality, you need to include the servers listed in [Basic Instant Messaging Deployment](#) and also install the Sun Java System Portal Server and Sun Java System Identity Server in your deployment.

In this deployment, use the directory and web servers used by the Identity Server. You do not need to install additional instances of those servers. In addition, because this deployment requires the Sun Java System Identity Server, all the features described in “Instant Messaging With Identity-based Server Policy Management or Single Sign On” on page 35 are also available.

Figure 2-7 shows the Portal-based deployment option.

**Figure 2-7** Instant Messaging With Portal-based Secure Mode or Archiving





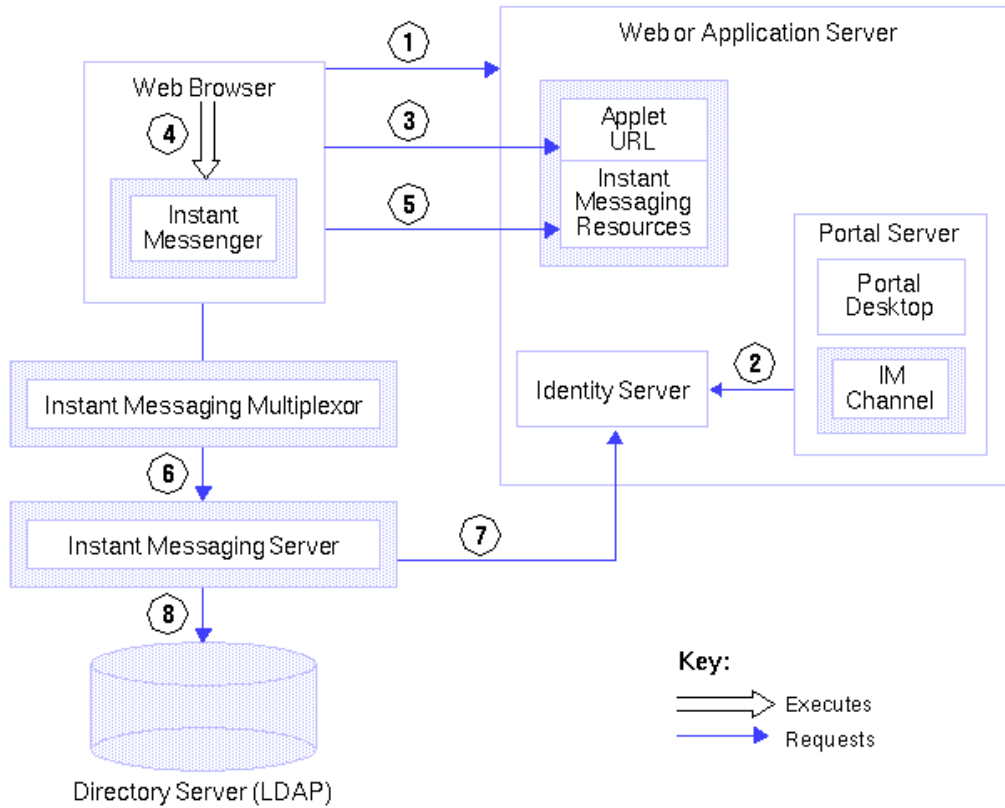
In this example:

- LDAP Server provides user entries.
- The Web Server (can also be an Application Server, which includes a Web Server) downloads the IM resources via a browser to the clients. The Resources are basically the client.
- Clients are always connect through a multiplexor.
- The Web Server (can also be an Application Server, which includes a Web Server) downloads the Instant Messaging resources via a browser to the clients. The Resources are basically the client.
- Clients are always connect through a multiplexor.
- Instant Messaging related services provided by the Identity server include the presence service and the Instant Messaging service.
- The Web server can be used to access the Identity server administration interface which manages the identity-based services for the Instant Messaging deployment. The web server for both Identity and Portal servers can be the same as the one that serves the Instant Messaging resources. See the Sun Java System Identity Server and Sun Java System Portal Server documentation for more details.
- Identity Server SDK provides the API used by the Instant Messaging server to interact with the Identity server.
- The Instant Messaging Channel is supported by the Portal server and allows users to access Instant Messenger from Portal Desktop.
- Portal Server provides archive functionality that allows you to save instant messages sent through the deployment.

## Authentication in a Deployment With Portal Server

[Figure 2-8](#) illustrates authentication process used by the Instant Messaging software in collaboration with the Sun Java System Portal Server and Sun Java System Identity Server components in a Single Sign-On environment. As with [Figure 2-2 on page 30](#), this figure focuses on the flow of authentication requests. An explanation of the steps in this process follows the figure.

**Figure 2-8** Flow of Authentication Requests in a Portal Server and Identity Server Configuration.



The authentication process of the Instant Messaging server in this deployment within a single sign-on environment works as follows:

1. The end user logs in to the Sun Java System Portal Server by entering the URL in a web browser.
2. The Sun Java System Identity Server software authenticates the end user and returns a session token and the Sun Java System Portal Server downloads the Desktop for the end user. The Sun Java System Portal Server Desktop is displayed in the end user's browser. See [Step 6](#) for an explanation of the session token.
3. The end user clicks the Sun Java™ System Instant Messenger URL link from the Instant Messaging channel on the Desktop.
4. The browser invokes Java Web Start or the Java Plug-in.
5. Java Web Start or the Java plug-in downloads the necessary Sun Java™ System Instant Messenger resource files and starts the Instant Messenger.
6. Instant Messenger requests authentication to the Instant Messaging server using the session token.

The session token is what enables single sign-on to work. This token is provided as an applet parameter and is used throughout the authentication process. End users are not asked for their credentials again as long as the session token is present.

7. The Instant Messaging server asks Sun Java System Identity Server to validate the session token. If the session is valid, Instant Messenger displays the end user's contact list and the end user can then use Instant Messenger services: chat, alerts, polls, etc.
8. The Instant Messaging server must query LDAP directly to get or set end-user information, such as contact lists or subscriptions.

# Instant Messaging With All Features Enabled

You can install Instant Messaging and enable all the features listed in this section. In order to do so, you must use the Sun Java™ System Messaging, Web, and Directory servers. In addition:

- You will need to install the following servers before you install Instant Messaging:
  - Sun Java System Directory Server (during Identity Server installation)
  - Sun Java System Web Server (during Identity Server installation)
  - Sun Java System Identity Server
  - Sun Java System Portal Server
  - Sun Java System Calendar Server
  - Sun Java System Messaging Server
- You need to install Instant Messaging resources on the web server's host.
- You need to install Sun Java System Identity Server SDK on the Instant Messaging server's host.
- You need to install the Identity Server Instant Messaging Service on the Identity Server's host.

## Physical Deployment Examples

Consider the following variations to the basic deployment scenario described earlier. For example, you can install the various required servers and components in the following physical configurations:

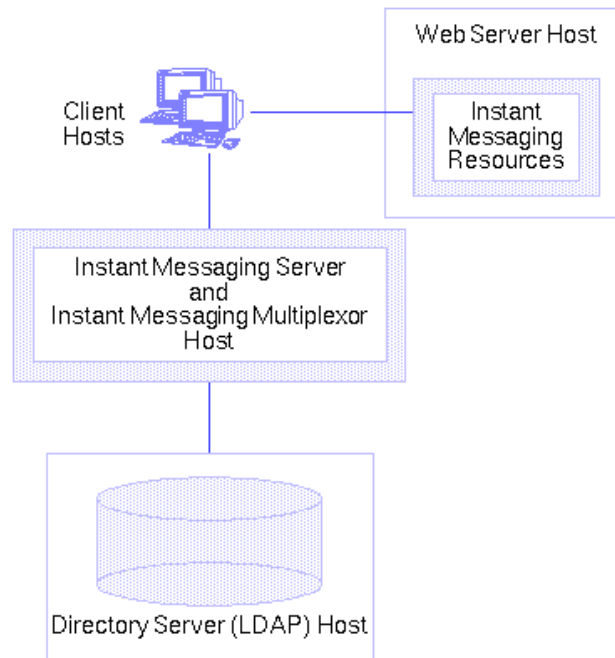
- [Instant Messaging Server and Web Server on Separate Hosts](#)
- [Multiple Multiplexors on Separate Hosts](#)
- [Federation of Multiple Instant Messaging Deployments](#)
- Any combination or all of the above

These variations can be applied in any of the deployment scenarios described in this chapter. You may choose to include them based on your deployment requirements.

## Instant Messaging Server and Web Server on Separate Hosts

Figure 2-9 shows a configuration where the Instant Messaging server and multiplexor are installed on the same host, and the web server is installed on a separate host. The Instant Messaging resources are also present on the web server host. Use this configuration when there is an existing instance of a web server and an LDAP server, and you do not want to install other applications on these hosts.

**Figure 2-9** The Web Server and Instant Messaging Installed on Separate Hosts.



## Multiple Multiplexors on Separate Hosts

Figure 2-10 shows a configuration of two multiplexors installed on separate hosts, and the Instant Messaging server on a different host. This configuration enables you to place a multiplexor outside your company's firewall. Installing multiplexors on multiple hosts distributes the load of the Instant Messaging server across multiple systems.

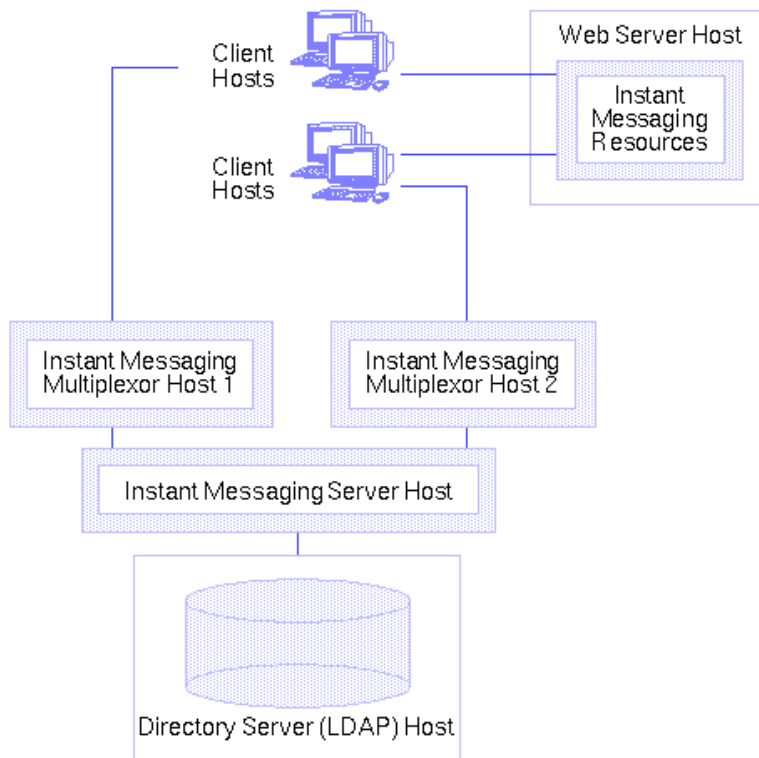
---

**NOTE** The multiplexor can be resource-intensive, so putting it on a separate host can improve the overall performance of the system.

Windows supports only one multiplexor instance per host.

---

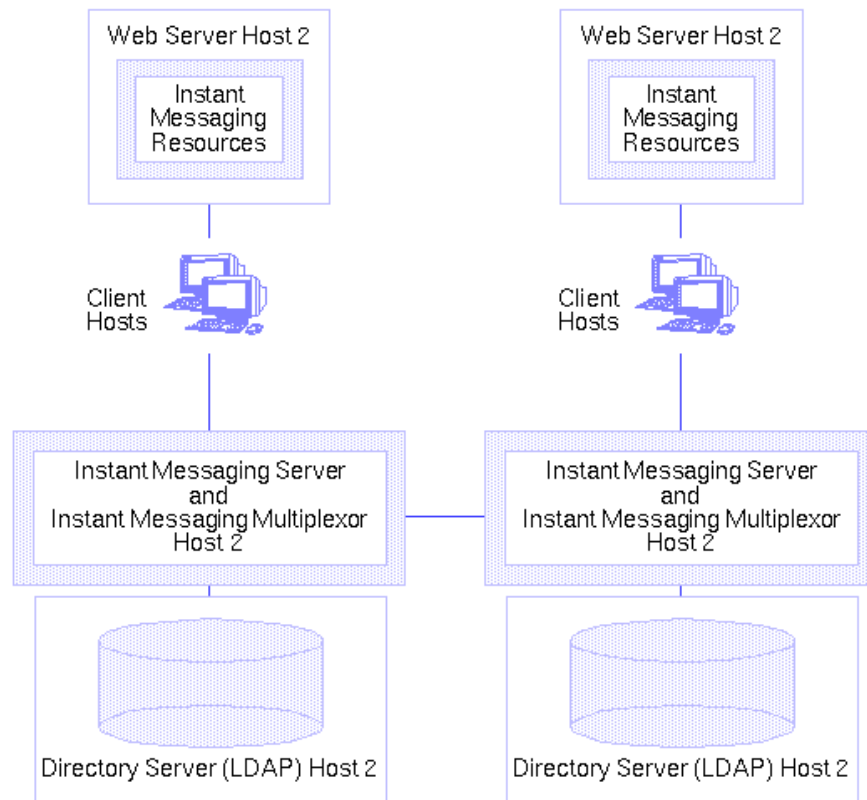
**Figure 2-10** Instant Messaging Multiplexors Installed on Separate Hosts.



## Federation of Multiple Instant Messaging Deployments

Figure 2-11 shows a configuration consisting of two Instant Messaging servers. This configuration is used when the site contains multiple administrative domains. The server configuration on each Instant Messaging server host has to be set up so that end users on one Instant Messaging server can communicate with end users on other Instant Messaging servers.

**Figure 2-11** Multiple Instant Messaging Server Hosts.



## Physical Deployment Examples



# Planning Your Sizing Strategy

When you design your deployment, you must decide how to configure your Instant Messaging server to provide optimum performance, scalability, and reliability.

Sizing is an important part of this effort. The sizing process allows you to identify what hardware and software resources are needed so that you can deliver your desired level of service or response time according to the estimated workload that your Instant Messaging server users generate. It is an iterative effort.

This chapter introduces the basics of sizing your Instant Messaging deployment to enable you to obtain the right sizing data by which you can make deployment decisions. It also provides the context and rationale for the Instant Messaging sizing process.

Since each deployment has its own set of unique features, this chapter will not provide detailed Instant Messaging sizing information for your specific site or sizing information for servers with which Instant Messaging interoperates, such as LDAP, SMTP, etc. Rather, this chapter explains what you need to consider when you architect your sizing plan, and provides general guidelines specifically for Instant Messaging components you can modify to suit your site's needs. Work with your Sun ONE technical representative for your deployment hardware and software needs.

The sizing process is outlined in the following topics:

- [Collecting Sizing Data](#)
- [Using a Load Simulator](#)
- [System Performance Guidelines](#)
- [Developing Architectural Strategies](#)
- [Example Resource Requirements](#)

# Collecting Sizing Data

Use this section to identify the data you need in order to size your Instant Messaging deployment. The following topics are covered in this section:

- [Determine Peak Volume of Unique Logins](#)
- [Create Your Usage Profile](#)
- [Define Your User Base or Site Profile](#)

## Determine Peak Volume of Unique Logins

Your *peak volume* is the largest concentrated numbers of unique logins to your Instant Messaging system within a given period in a day. The volume can vary from site to site as well as across different classes of users. For example, peak volume among groups may occur during corporate-held core hours which differ between time zones.

Consider the following points when analyzing peak volume:

1. Determine when and for how long the peaks occur.
2. Size your deployment against peak volume load assumptions.

Once patterns are analyzed, choices can be made to help the system handle the load and provide the services that users demand.

3. When you determine the peak volume on your system, be sure that your Instant Messaging deployment can support it.

## Create Your Usage Profile

Measuring your load is important for accurate sizing. Your *usage profile* determines the factors that programs and processes place on your Instant Messaging servers and multiplexors.

This section helps you create your *usage profile* to measure the amount of load that is placed on your deployment.

To create a usage profile, answer the following questions:

1. What is the total number of users on your system?

When counting the number of users on your system, account for not only the users who have accounts and can log into the system, but also the users with accounts who are currently not logged into the system. [Table 3-1](#) describes the types of users that make up the total.

Characterize your configured users using these three general profiles. The total of these users should give you an idea of the total number of concurrent connections you need to support.

**Table 3-1** Active Versus Inactive User

Connection	Description
Inactive User	A user with an Instant Messaging account who currently is not logged into the system. Non-connected users consume disk space but no CPU or memory.
Connected/Inactive	These users are logged in, but are not currently sending or receiving instant messages.
Connected/Active	Logged into the system and actively sending messages, updating user information such as contact lists, and attending conferences throughout the day.

If you have a small deployment, the defaults may be sufficient to meet your site's needs. So, if you have a very small deployment (for example: under 300 users), you may not need to go through this process of planning a sizing strategy. Work with your Professional Services representative to determine your individual needs.

2. How many connections are on your system during your peak volume?

Correctly formulating the maximum number of concurrent users that has to be sustained by the system is key to planning your resource requirements. Although a deployment usually has maximum number of configured users, it is important to plan for the maximum number of concurrent users (connected and more or less active). A conservative estimate for the number of concurrent users can then be determined based on a 1:10 ratio. Thus, for a deployment of 50,000 configured users, the concurrent users would be 5,000.

Specifically, note the number of concurrent connections, idle, and busy connections.

**Table 3-2** Client Connections

Connection	Description
Concurrent Connection	Number of unique TCP connections or sessions that are established on your system at any given time.
Idle Connection	Connection where no information is being sent between the client and multiplexor or server and multiplexor.
Busy Connection	A connection that is in progress. An established connection where information is being sent between the client and multiplexor or multiplexor and server.

To determine the number of *concurrent connections* in your deployment, you can either:

- a. Count the number of established TCP connections by using the `netstat` command on Unix platforms.
- b. Obtain the last login and logout times for Instant Messenger users.

To determine the number of concurrent connections you can support, you need to obtain two values from parameters within the `iim.conf` file that are used for tuning multiplexor performance:

- a. `iim_mux.numinstances` - Specifies the number of multiplexor instances.
- b. `iim_mux.maxsessions` - Specifies the maximum number of clients that one mutliplexor process can handle. The default is 1000.

Once you have obtained these values, multiply the `numinstances` number by the `maxsessions` number. This gives the total number of concurrent connections supported by your deployment. For information on locating the `iim.conf` file, see the *Sun Java System Instant Messaging Administration Guide*.

3. If you have a large deployment, how will you organize your users?

For example, consider placing active users and inactive users together on separate machines from one another.

If an inactive user becomes an active user, that user can be moved to the active user machines. This approach could decrease the amount of needed hardware, rather than placing inactive and active users together on a machine.

4. What is the amount of storage used for each user?

If you are not storing your end user data, such as contact lists, in LDAP, you need to plan for the space required to store this data. If you configure the server to store this data outside LDAP, the server stores it in a flat file. See the *Sun Java System Instant Messaging Administration Guide* for more information.

5. How many messages enter your Instant Messaging system from the Internet?

The number of messages should be measured in messages per second during your peak volume.

6. How many messages are sent by your users to:

- o end users on your system?
- o the Internet?

This number of messages is also measured in messages per second during the peak volume.

7. Will you be using Secure Shared Layer (SSL)? If yes, what percentage of users and what type of users?

For example, in a particular organization, 20% of connections during peak hours will enable SSL.

Answering these questions provides a preliminary usage profile for your deployment. You can refine your usage profile as your Instant Messaging needs change.

### Additional Questions

While the following questions are not applicable to creating your usage profile, they are important to developing your sizing strategy. How you answer these questions may require you to consider additional hardware.

1. How much redundancy do you want in your deployment?

For example, do you need to consider high availability.

2. What backup and restore strategy do you have in place (such as disaster recovery and site failover)? What are the expected times to accomplish recovery tasks?

Typically you need to back up the server configuration files, database, and any resource files you have customized.

## Define Your User Base or Site Profile

Once you establish a usage profile, compare it to sample pre-defined user bases that are described in this section. A *user base* is made up of the types of Instant Messaging operations that your users will perform. Instant Messaging users fall into one of these user bases:

- [Casual Users](#)
- [Heavy Users](#)

The sample user bases described in this section broadly generalize user behavior. Your particular usage profile may not exactly match the user bases; you will be able to adjust these differences when you run your load simulator (as described in [Using a Load Simulator](#)).

### Casual Users

A lightweight user base typically consists of users with simple Instant Messaging requirements. These users rarely initiate chat sessions and rarely receive invitations. They may only use Instant Messaging as a presence tool.

### Heavy Users

A heavy user uses significantly more system resources than a casual user. Typical usage for a this type of user may be something like the following:

- Presence updates equal to or greater than 20 times a day.
- Contact list contains about 30 contacts.
- Subscribes to the presence updates of all the contacts in the contact list.
- Sets up around 4 conferences or chats per day where each conference has 3 people in the conference room and lasts an average of 10 minutes, and a message is added to the conference every 1 -15 seconds.

# Using a Load Simulator

To measure the performance of your Instant Messaging architecture, use your user bases (described in “[Define Your User Base or Site Profile](#)”) and your usage profile (described in “[Create Your Usage Profile](#)”) as inputs into a load simulator.

A load simulator creates a peak volume environment and calibrates the amount of load placed on your servers. You can determine if you need to alter your hardware, throughput, or deployment architecture to meet your expected response time, without overloading your system. To use a load simulator:

1. Define the user base that you want to test (for example, casual users).  
If necessary, adjust individual parameters to best match your usage profile.
2. Define the hardware that will be tested.
3. Run the load simulator and measure the maximum number of concurrent connections on the tested hardware with the user base.
4. Publish your results and compare those results with production deployments.
5. Repeat this process using different user bases and hardware until you get the response time that is within an acceptable range for your organization under peak load conditions.

---

**NOTE** Contact Sun Professional Services for recommended load simulators and support.

---

## System Performance Guidelines

Once you evaluate your hardware and user base with a load simulator, you need to assess your system performance. The following topics address methods by which you can improve your overall system performance:

- [Memory Utilization](#)
- [Disk Throughput](#)
- [Disk Capacity](#)
- [Network Throughput](#)
- [CPU Resources](#)
- [Multiplexor Configuration Best Practices](#)

## Memory Utilization

Make sure you have an adequate amount of physical memory on each machine in your deployment. Additional physical memory improves performance and enables the server to operate at peak volume. With sufficient memory, Instant Messaging can operate efficiently without excessive swapping.

For most deployments, you need at least 256 MB of RAM. The amount of RAM needed depends on the number of concurrent client connections, and whether the server and multiplexor are deployed on the same host. For information about concurrent connections, see [“Create Your Usage Profile” on page 50](#). For information on hosting the server and multiplexor on the same host, see [“Developing Architectural Strategies” on page 59](#).

On Unix, you can set the amount of memory allocated to the server by modifying the `iim.jvm.maxmemorysize` parameter in the `iim.conf` file. This parameter specifies the maximum number of megabytes of memory that the JVM running the server is allowed to use. The default setting is 256 MB, and the maximum setting is 500 MB. For instructions on modifying this parameter, see the *Sun Java System Instant Messaging Administration Guide*. You cannot currently change this value on Windows NT.

## Disk Throughput

Disk throughput is the amount of data that your system can transfer from memory to disk and from disk to memory. The rate at which this data can be transferred is critical to the performance of Instant Messaging. To improve efficiency in your system's disk throughput:

- Consider your maintenance operations, and ensure you have enough bandwidth for backup. Backups can also affect network bandwidth, particularly remote backups. Private backup networks can be a more efficient alternative.
- Carefully partition the data stores to improve throughput efficiency.
- Stripe data across multiple disk spindles in order to speed up operations that retrieve data from disk.



## Disk Capacity

When planning server system disk space, you need to be sure to include space for operating environment software, Instant Messaging software, and for any servers not currently in your network that need to be installed to support Instant Messaging (such as LDAP). Be sure to use an external disk array. In addition, user disk space needs to be allocated. Typically, this space is determined by your site's policy. Typical installations will require:

- Approximately 300 MB of free disk space for each server or multiplexor.
- Approximately 5K of disk space for each user.
- Additional space for the Instant Messaging archive.

The archive captures instant messages and archives them in a Portal Server Search database. End users can query and retrieve these archived messages using the Search page on the Portal Server desktop.

Use [Table 3](#) to determine server and multiplexor disk space sizing numbers whether archiving is enabled or disabled. The figures listed in the table were generated using a 400MHz Ultra Sparc II Processor.

**Table 3** Server and Multiplexor Memory Disk Space Sizing for Concurrent Users

	<b>Server Memory Consumption for Connected/Inactive Users</b>	<b>Server Memory Consumption for Connected/Active Users</b>	<b>Multiplexor Memory Consumption for Connected/Inactive Users</b>	<b>Multiplexor Memory Consumption for Connected/Active Users</b>
Archive Disabled	8 MB +20 K per User	120 MB + 20 K per User	8 MB + 20 K per User	8MB + 28K per User
SSO/Portal/Archive enabled	100MB +25K per User	120MB +30K per User	8M+35K per user	8 MB +40K per user

Please refer to the product documentation for other servers interoperating with Instant Messaging for their specific requirements.

## Network Throughput

Network throughput is the amount of data at a given time that can travel through your network between your client application and server.

- To avoid bottlenecks, ensure that the network infrastructure can handle the load.
- Partition your network.
- To ensure that sufficient capacity exists for future expansion, don't use theoretical maximum values when configuring your network.
- Separate traffic flows on different network partitions to reduce collisions and to optimize bandwidth use.

## CPU Resources

Enable enough CPU for your servers and multiplexing services. In addition, enable enough CPU for any RAID systems that you plan to use. If you intend to use archiving in your deployment, you need to take those space requirements into consideration as well.

Use [Table 4](#) to help determine the number of CPUs your installation requires for optimum performance whether archive is enabled or disabled. The figures listed in the table were generated using a 400MHz Ultra Sparc II Processor.

**Table 4** CPU Utilization Numbers

	<b>Server CPU Utilization for Connected/Inactive Users</b>	<b>Server CPU Utilization for Connected/Active Users</b>	<b>Multiplexor CPU Utilization for Connected/Inactive Users</b>	<b>Multiplexor CPU Utilization for Connected/Active Users</b>
Archive Disabled	Several hundred thousand users per CPU	30 K users per CPU	50 K users per CPU	5 K users per CPU

## Multiplexor Configuration Best Practices

Consider the following suggestions and generalizations when planning your multiplexor deployment. The parameters discussed in this section are located in the `iim.conf` file. See the *Sun Java System Instant Messaging Administration Guide* for more detailed information about these parameters.

- The number of `iim_mux.maxthreads` should not exceed the number of CPUs on your server.

This helps maximize resource utilization and optimizes processing speed.

- The `im_mux.maxsessions` value should be high enough to avoid rejecting connections, but it should be reasonable enough so that the multiplexor processes do not get overloaded.
- Be sure that your expected number of concurrent client connections is less than the maximum possible by a safe margin.
- Do not configure threads or number of concurrent sessions to more than you require. Otherwise, you will unnecessarily consume system resources.
- A good starting point is to configure `im_mux.numinstances` to the number of CPUs on the system.

## Developing Architectural Strategies

Once you have identified your system performance needs, the next step in sizing your Instant Messaging deployment is to size specific components based on your architectural decisions.

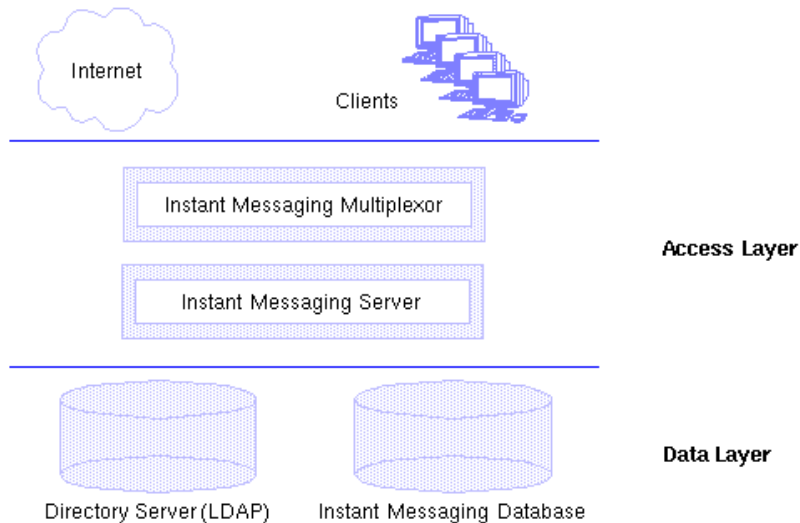
The following sections point out sizing considerations when you deploy two-tiered and one-tiered architectures:

- [Two-tiered Architecture](#)
- [One-tiered Architecture](#)

### Two-tiered Architecture

A two-tiered architecture splits the Instant Messaging server deployment into two layers: an access layer and a data layer. In a simplified two-tiered deployment, you might add one or more multiplexors and servers to the access layer. The multiplexor acts as a proxy for users, and relays messages to the Instant Messaging server. The data layer holds the Instant Messaging server database and Directory servers. [Figure 3-1](#) shows a simplified two-tiered architecture.

**Figure 3-1** Simplified Two-tiered Architecture



Two-tiered architectures have advantages over one-tiered architectures that can impact your sizing decisions. Two-tiered architectures:

- Are easier to maintain than one-tiered architectures.
- Allow the offloading of load-intensive processes like SSL, message reprocessing.
- Are easier for growth management and you can upgrade your system with limited overall downtime.

### To Size Your Multiplexing Services

When you size your multiplexor, the calculation is based on your system load, particularly the number of concurrent connections the multiplexor needs to handle.

In addition, you must:

1. Add CPU or a hardware accelerator for SSL if appropriate.
2. Add memory to the machine if the multiplexor is being configured on it.
3. Account for Denial of Service.

4. Add capacity for load balancing and redundancy, if appropriate.

One or more of each type of machine should still handle peak load without a substantial impact to throughput or response time when you plan for redundancy in your deployment.

## One-tiered Architecture

In a one-tiered architecture, there is no separation between access and data layers. The Instant Messaging server, multiplexor, and sometimes the Directory server are installed in one layer. [Figure 3-2](#) illustrates the idea.

**Figure 3-2** Simplified One-tiered Architecture



One-tiered architectures have lower up-front hardware costs than two-tiered architectures. However, if you choose a one-tier architecture, you need to allow for significant maintenance windows.

To size a one-tiered architecture:

1. Add CPU for SSL, if necessary.
2. Account for Denial of Service attacks.
3. Add more disks for the increased number of client connections.
4. Add more disks for each multiplexor.

For specific instructions on sizing Instant Messaging components in one-tiered or two-tiered architectures, contact your Professional Services representative.

# Example Resource Requirements

This section provides example resource distributions and recommended sizing information for the following two deployment types:

- [Small Deployment Sample Resource Requirements Numbers](#)
- [Large Deployment Sample Resource Requirements Numbers](#)

## Small Deployment Sample Resource Requirements Numbers

For a small deployment with the server and multiplexor on a single server having 10,000 users with the following profile:

- 30% connected/active
- 20% connected/inactive
- 50% not connected

The memory requirements are: 1 or 2 CPUs with 300-500 MB RAM each.

## Large Deployment Sample Resource Requirements Numbers

For a large deployment having 1,000,000 users with the following profile:

- 5% connected/active
- 20% connected/inactive
- 75% not connected

The server memory requirements are 4 GB RAM on 2 CPUs. The multiplexor requirement is 4 GB RAM on 16 CPUs.

## SYMBOLS

[ and ] 8

## A

access protocols, LDAP 17  
alerts 22  
API, for Instant Messaging 13  
applet URL 31  
architectural strategies 59  
architecture  
  basic 19  
  one-tiered 61  
  two-tiered 59  
authentication  
  Identity server and 37  
  LDAP and 15  
  Portal server and 41  
  users and 15  
authentication, and basic deployment 29

## B

basic architecture 19  
basic deployment  
  authentication and 29

## C

calendar alerts 22  
calendar event notification, *See event notification* 15  
Calendar server 15  
  requirements 15  
casual users 54  
chat 22  
client resources, *See Instant Messenger resources* 14  
client, *See Instant Messenger* 12  
cn attribute 15  
collecting sizing data 50  
communication modes 22  
communication protocols  
  HTTP 18  
  TCP/IP 18  
components  
  core 12–13  
  related 14–16  
conference rooms 22  
conventions 7–8  
  [ and ] 8  
  bold monospaced font 7  
  command-line prompts 8  
  italicized font 7  
  monospaced font 7  
  square brackets 8  
  straight brackets 8  
core components 12–13  
  Instant Messaging multiplexor 21  
  Instant Messaging server 20  
  Instant Messenger 22

CPU resources 58

## D

### defining

- peak volume 50
- site profile 54
- usage profile 50
- user base 54

### deployment

- authentication and 37, 41
  - calendar event notification and 33
  - Calendar server and 33
  - design overview 23
  - Directory server and 29
  - email notification and 31
  - examples 27–47
  - Identity server and 35
  - multiple Instant Messaging servers 47
  - physical 44
  - piloting 25
  - planning process 23
  - Portal server and 39
  - production and 25
  - resources for large 62
  - resources for small 62
  - SMTP server and 31
  - Web server and 29
- Directory server, requirements 15
- directory service, defined 13
- disk capacity 57
- disk throughput 56
- document conventions, *See conventions* 7
- documentation, online 8

## E

### email

- Instant Messaging and 15
  - notification 31
- embedded URLs 23

event notification 15, 33

### examples

- deployment 27–47
- resource requirements 62

## F

### files

- im.conf 56
- imbrand.jar 14
- imdesktop.jar 14
- imjini.jar 14
- imnet.jar 14
- imres.jar 14
- index.html 14

## G

### guidelines

- multiplexor 58
- performance 55

## H

- heavy users 54
- HTML, and instant messages 17
- HTTP 17
  - communication protocols 18
- hyperlinks, and instant messages 17

## I

- Identity server 15
- Identity Server SDK 15
- im.conf file 56
- im.jvm.maxmemorysize parameter 56
- im\_mux.maxsessions parameter 52, 59



- iim\_mux.maxthreads parameter 58
- iim\_mux.numinstances parameter 52, 59
- imbrand.jar file 14
- imdesktop.jar file 14
- imjini.jar file 14
- imnet.jar file 14
- imres.jar file 14
- index.html file 14
- instant message
  - archiving 39
  - embedded URLs in 23
  - flow 19
  - HTML and 17
  - hyperlinks in 17
  - structure 17
- Instant Messaging
  - API 13
  - Calendar server and 33
  - channel 16
  - core components 12–13
  - database 15
  - Directory server and 13
  - email and 15
  - Identity server and 35
  - Messaging server and 31
  - overview 11–25
  - Portal server and 39
  - protocols 13
  - related components 14–16
  - SMTP server and 31
- Instant Messaging multiplexor 12, 21
  - best practices 58
  - configuration 58
  - deploying multiple 46
  - resource requirements and 46
  - Windows support and 46
- Instant Messaging resources, *See Instant Messenger resources* 14
- Instant Messaging server
  - defined 12, 20
  - deploying multiple 47
  - message delivery 21
- instant messaging service 12
- Instant Messenger 12, 22
  - communication modes 22

- non-secure mode 16
- secure mode 16
- Instant Messenger resources
  - files 14
  - installing 14
  - Web server and 14

## L

- LDAP 13, 17
  - attributes for Instant Messaging 15
  - direct lookup 21
  - user attributes 15
  - user authentication and 15
- LDAP server, requirements 15
- load simulator, and deployment 55

## M

- memory
  - maximum for JVM 56
  - utilization 56
- message delivery, and Instant Messaging server 21
- message transfer protocols, SMTP 18
- Messaging server, requirements 15
- messenger.jar file 14
- multiplexor, *See Instant Messaging multiplexor* 12

## N

- network throughput 57
- news channels 22
- non-secure mode, Instant Messenger 16

## O

- offline message forwarding 15

one-tiered architecture 61

options

on Linux 27

on Solaris 27

on Windows 27

overview 11-25

## P

parameters

im.jvm.maxmemorysize 56

im\_mux.maxsessions 52, 59

im\_mux.maxthreads 58

im\_mux.numinstances 52, 59

peak volume

defined 50

defining 50

performance guidelines 55

physical deployment examples 44

piloting, deployment 25

planning

deployment 23

sizing 49-62

policy management 35

poll 22

Portal server 16

Portal Server Desktop 16

Secure Remote Access 16

prerequisites 6

protocols

access 17

communication 18

HTML 17

HTTP 17, 18

Instant Messaging and 13

LDAP 17

message transfer 18

SMTP 17, 18

TCP/IP 17, 18

## R

related components 14-16

requirements 15

Calendar server 15

Directory server 15

Instant Messenger resources 14

LDAP server 15

resource examples 62

SMTP server 15

Sun ONE Identity Server 15

Sun ONE Portal Server 16

Web server 14

resource requirements

CPU 58

large deployment 62

small deployment 62

resources, *See Instant Messenger resources* 14

## S

SDK

Identity Server 15

secure mode 39

Instant Messenger 16

server, *See Instant Messaging server* 12

single sign on 35

site profile 54

sizing

collecting data 50

planning 49-62

process 49

SMTP 17

message transfer protocol 18

SMTP server, requirements 15

standards, supported 17

structure, instant message 17

Sun ONE Application Server SE 14

Sun ONE Calendar Server, requirements 15

Sun ONE Identity Server 15

deployment and 35

Portal server and 39

requirements 15

- Sun ONE Identity Server Instant Messaging Service
  - Definition 13
- Sun ONE Identity Server SDK, requirements 15
- Sun ONE Portal Server 16
  - deployment and 39
  - Identity server and 39
  - requirements 16
- Sun ONE Portal Server Netlet 16
- Sun ONE Portal Server, Secure Remote Access 16
- supported standards 17
  - HTML 17
  - HTTP 17
  - LDAP 17
  - SMTP 17
  - TCP/IP 17

## T

- TCP/IP 17
  - communication protocol 18
- tuning
  - iim.conf settings for multiplexor 52
  - rules of thumb 58
- two-tiered architecture 59

## U

- uid attribute 15
- URL
  - applet 31
  - embedded 23
- usage profile
  - defined 50
  - defining 50
- user base
  - defined 54
- users
  - authenticating in LDAP 15
  - casual 54
  - heavy 54

## W

- Web server 14
  - installing on separate host 45
  - Instant Messenger resources and 14
  - requirements 14
- Windows
  - JVM memory setting and 56

